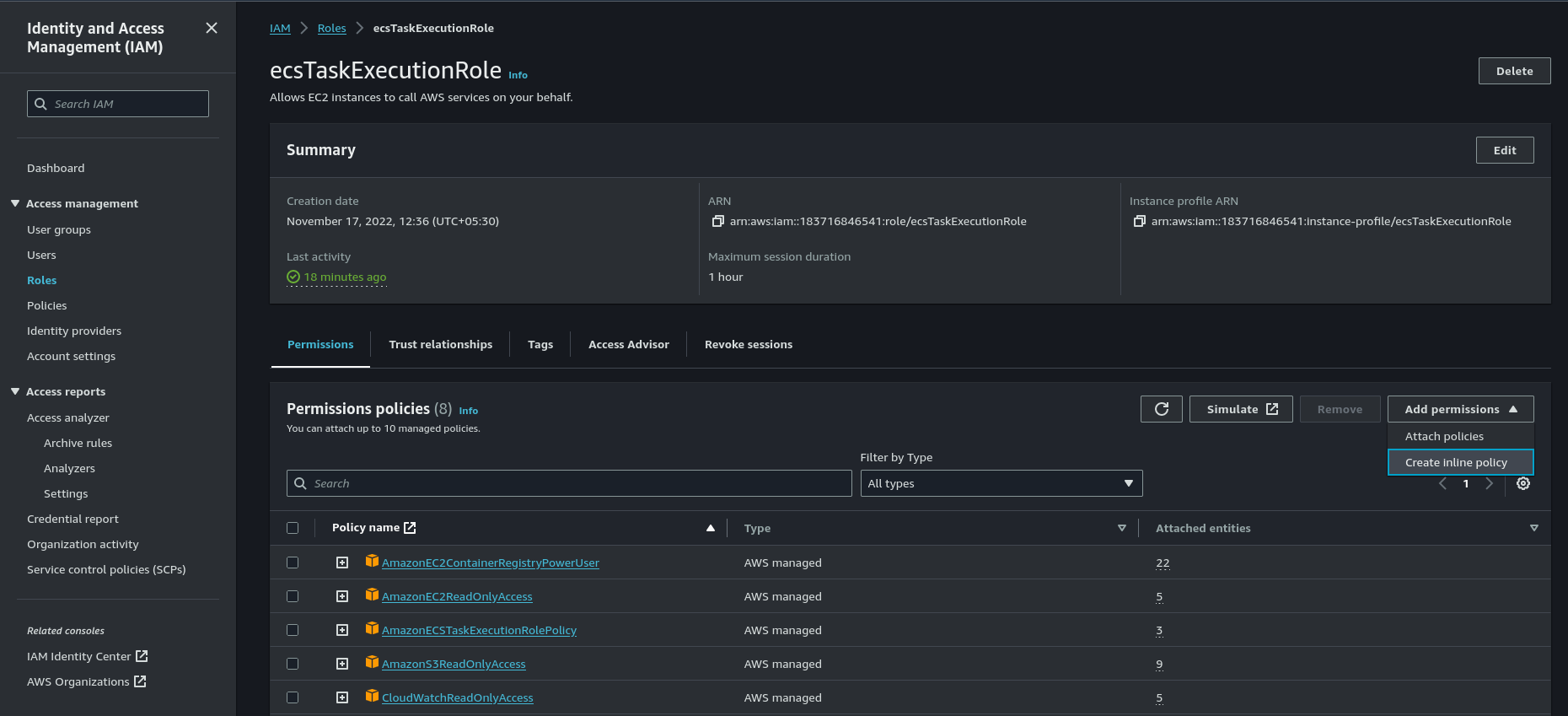
# Deploying Loki With S3 as backend storage in ECS.

## Requirements:

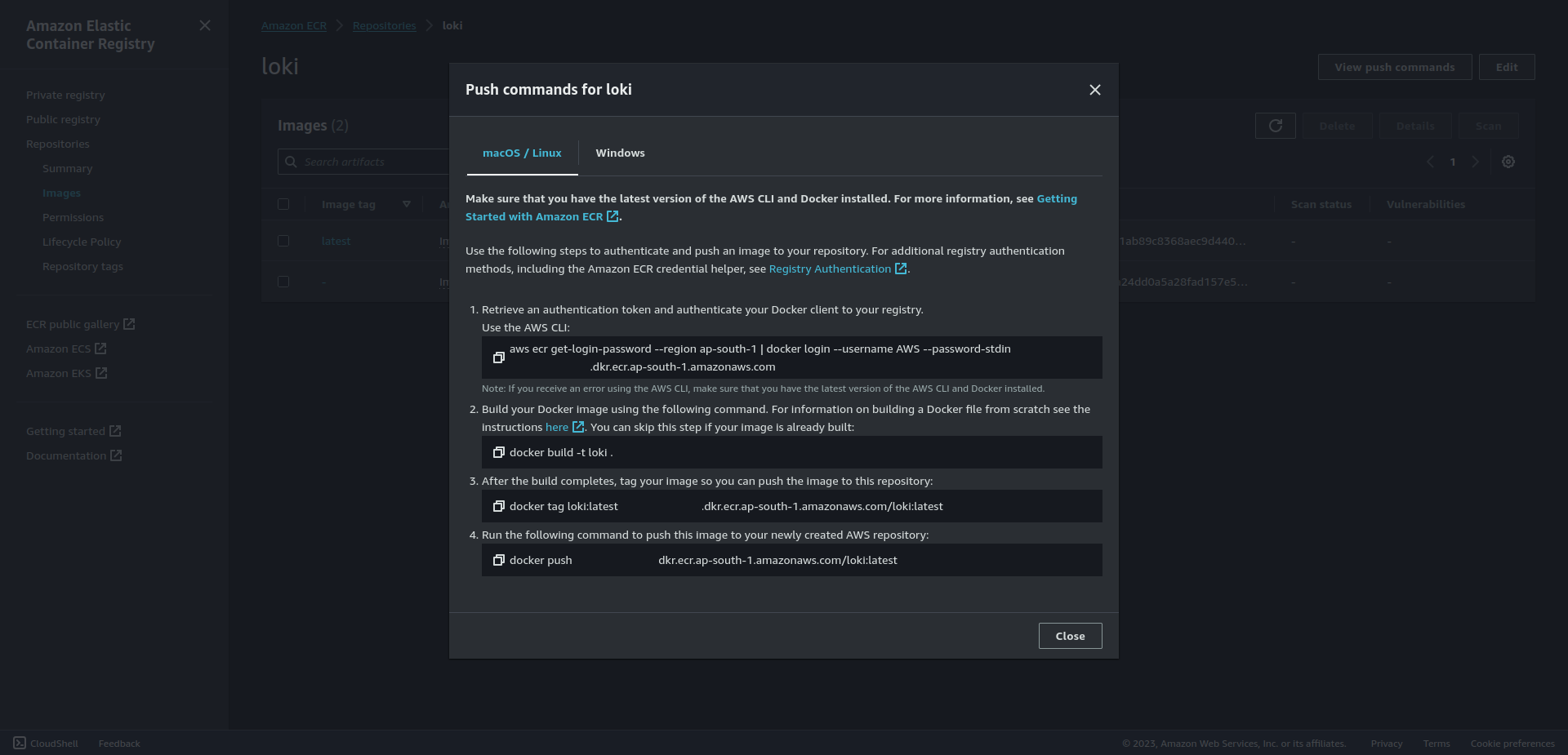
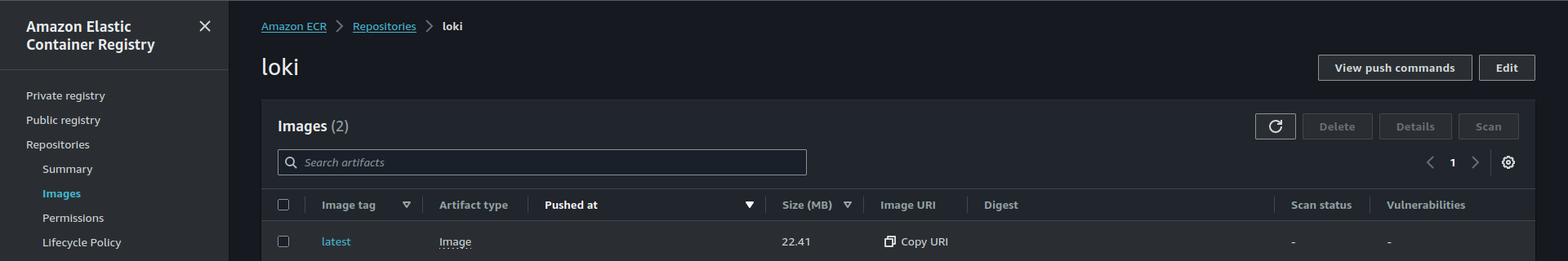
* AWS Account.
* Privileges to modify IAM roles.
* Working ECS Cluster.
* S3 bucket for Loki
* Grafana running. (Not a hard requirement\* but important for Integration)

## Steps:

### Create/Modify ECS task IAM role so that Loki in ECS can use S3 bucket.

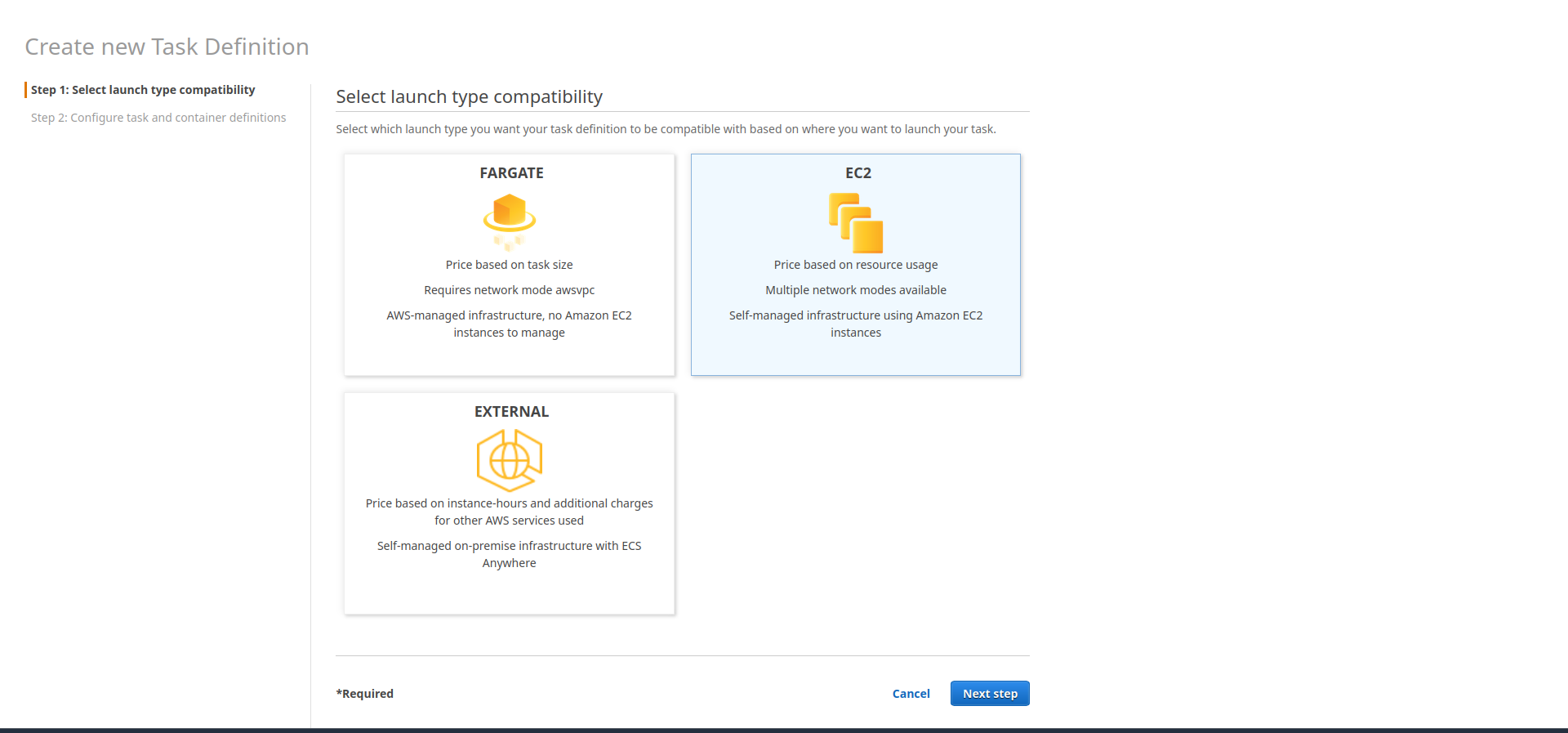
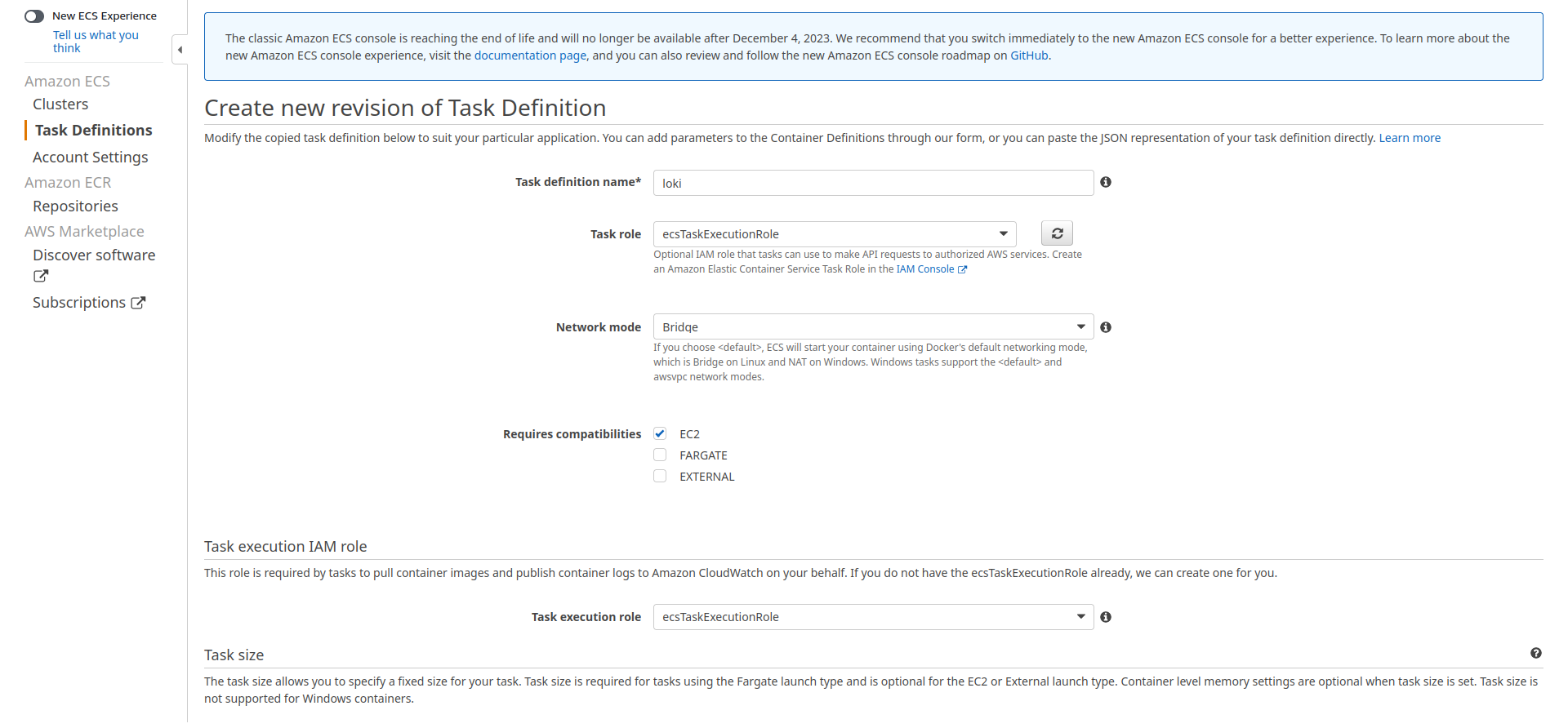
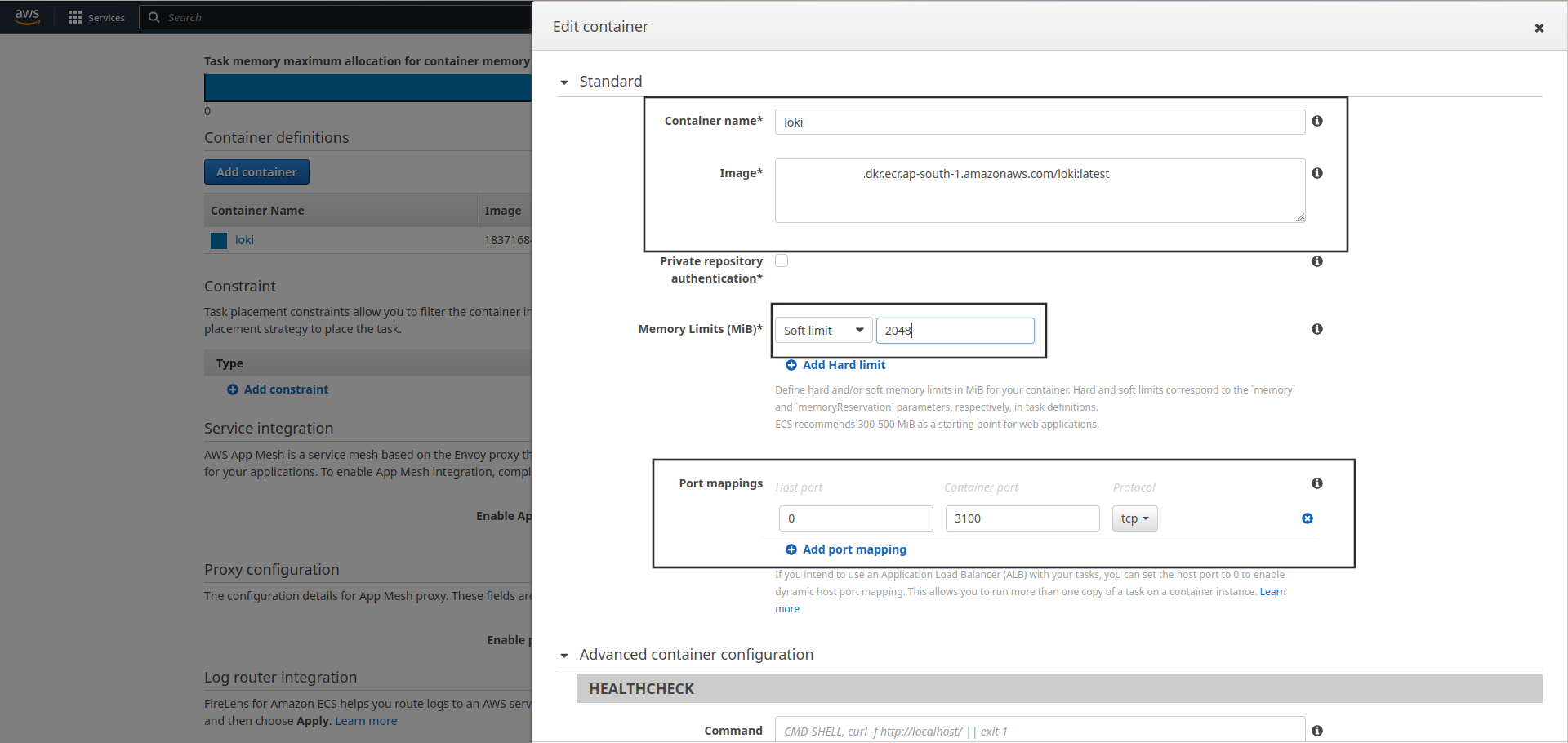
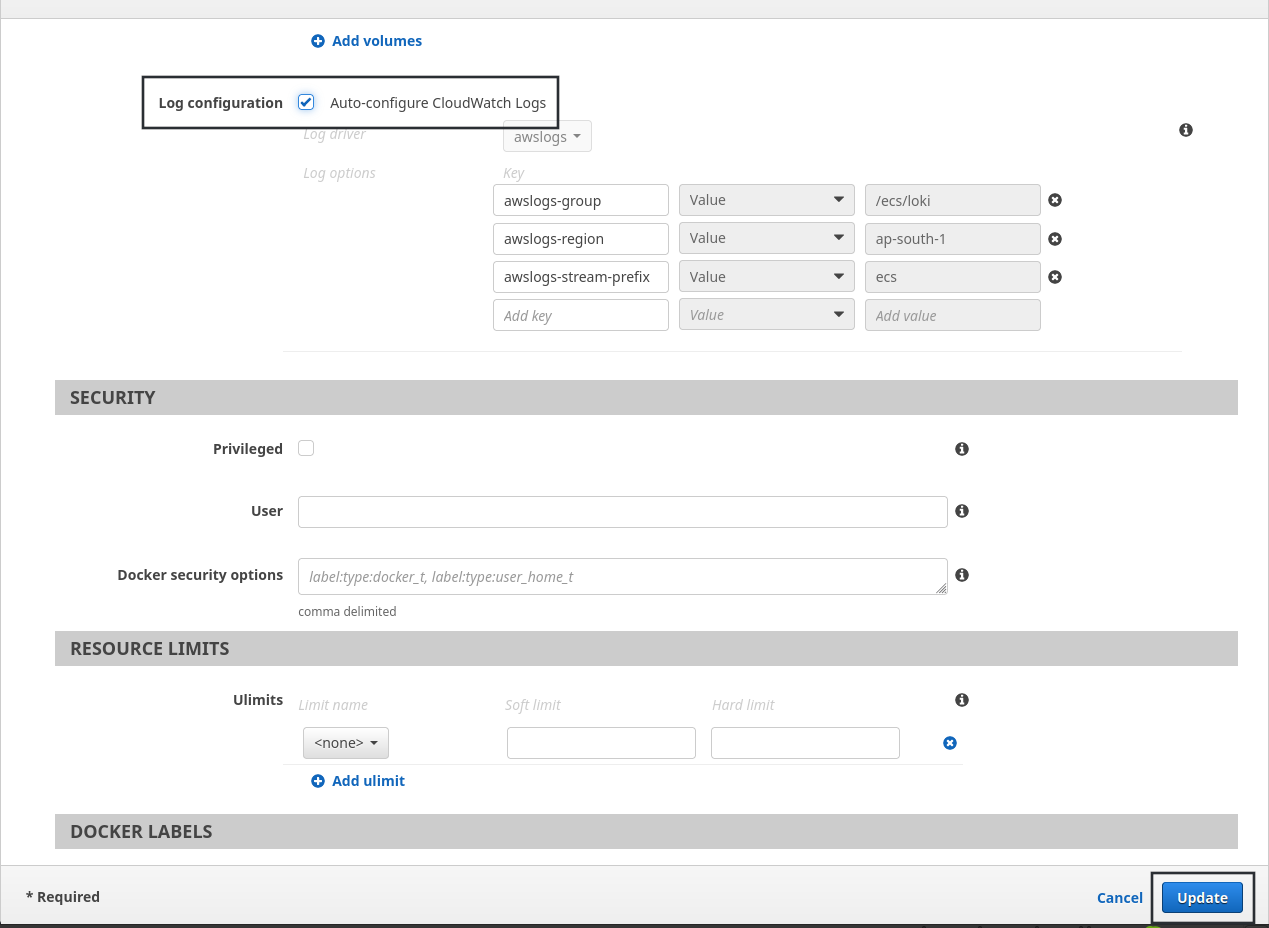
* Go to IAM -> select roles -> Search for ***ecsTaskExecutionRole*** -> Select that role -> Under **Permission Policies** select ***Add permissions*** -> Hit ***Create Inline Policy***
* 
* Select json and paste following policy with one minor adjustment which is in place of put your s3 bucket name.
* {  
  "Version": "2012-10-17",  
  "Statement": [  
   {  
   "Sid": "Statement1",  
   "Effect": "Allow",  
   "Action": [  
   "s3:ListBucket",  
   "s3:ListAllMyBuckets",  
   "s3:GetObject",  
   "s3:PutObject",  
   "s3:DeleteObject"  
   ],  
   "Resource": [  
   "arn:aws:s3:::<bucket\_name>",  
   "arn:aws:s3:::<bucket\_name>/\*"  
   ]  
   }  
  ]  
  }
* Hit ***next*** and then ***Create policy***

### Create Custom docker image for Loki and Uploading it to ECR.

* First Navigate to ECR and hit ***create repository*** (we will be creating private repo here.) -> Keep Visibility Setting ***Private*** -> Give it appropriate name and hit ***Create Repository***.
* Now In your local machine or in any Linux machine create 2 files one being ***Dockerfile*** and ***local-config.yaml*** and add following content.
* Dockefile docker FROM grafana/loki:latest COPY local-config.yaml /etc/loki/local-config.yaml
* local-config.yaml (change 2 values: , ) ```yaml auth\_enabled: false
* server: http\_listen\_port: 3100 grpc\_listen\_port: 9096
* ingester: wal: enabled: true dir: /tmp/wal lifecycler: address: 127.0.0.1 ring: kvstore: store: inmemory replication\_factor: 1 final\_sleep: 0s chunk\_idle\_period: 1h # Any chunk not receiving new logs in this time will be flushed max\_chunk\_age: 1h # All chunks will be flushed when they hit this age, default is 1h #chunk\_target\_size: 1048576 # Loki will attempt to build chunks up to 1.5MB, flushing first if chunk\_idle\_period or max\_chunk\_age is reached first chunk\_target\_size: 209715200 chunk\_retain\_period: 30s # Must be greater than index read cache TTL if using an index cache (Default index read cache TTL is 5m) max\_transfer\_retries: 0 # Chunk transfers disabled
* schema\_config: configs: - from: 2020-10-24 store: boltdb-shipper object\_store: s3 schema: v11 index: prefix: index\_ period: 24h
* storage\_config: boltdb\_shipper: active\_index\_directory: /tmp/loki/boltdb-shipper-active cache\_location: /tmp/loki/boltdb-shipper-cache cache\_ttl: 24h # Can be increased for faster performance over longer query periods, uses more disk space resync\_interval: 5s shared\_store: s3 aws: # Change following 2 values region: bucketnames: s3forcepathstyle: true
* compactor: working\_directory: /tmp/loki/boltdb-shipper-compactor shared\_store: filesystem
* limits\_config: retention\_period: 72h enforce\_metric\_name: false reject\_old\_samples: true reject\_old\_samples\_max\_age: 168h max\_cache\_freshness\_per\_query: 10m split\_queries\_by\_interval: 15m # for big logs tune per\_stream\_rate\_limit: 512M per\_stream\_rate\_limit\_burst: 1024M cardinality\_limit: 200000 ingestion\_burst\_size\_mb: 1000 ingestion\_rate\_mb: 10000 max\_entries\_limit\_per\_query: 1000000 max\_label\_value\_length: 20480 max\_label\_name\_length: 10240 max\_label\_names\_per\_series: 300 max\_query\_series: 100000
* chunk\_store\_config: max\_look\_back\_period: 0s
* table\_manager: retention\_deletes\_enabled: false retention\_period: 0s
* ruler: storage: type: local local: directory: /tmp/loki/rules rule\_path: /loki/rules-temp alertmanager\_url: http://localhost:9093 ring: kvstore: store: inmemory enable\_api: true
* query\_scheduler: max\_outstanding\_requests\_per\_tenant: 2048 ```
* Now build the docker image with the help of ECR for that go to the ECR repo and hit ***View push commands*** and follow the instructions.
* 
* Check if the Image is pushed properly or not.
* 

### Deploy Loki in ECS.

#### Create Task definition:

* Refer [Grafana deployment](01_Deploying_Grafana.md) for ECS task definition configuration (ignore EFS setup in that).
* Navigate to ECS -> On left hand pane Select Task Definitions -> Create New task Definition
* Under Select launch type compatibility select appropriate Launch type, Assuming that we are using ASG (Auto Scaling Group as Capacity Provider in the ECS cluster) we will select EC2 -> Hit Next Step.
* 
* In the next step we will Configure task and container definitions.
* In the form add appropriate information. Name -> Task Role -> In network mode select Bridge Mode -> Task Execution Role (use default role or create a new IAM role and use that; but make sure if you are using new IAM role select Configure task and container definitions as use case under Elastic Container Service)
* 
* After this scroll down and under Container Definitions select Add Container -> A new form will appear -> here give appropriate Container name -> Add docker image name -> in Memory Limits select Soft limit and set memory accordingly for this demo we will set it as 2048 which is 2GiB -> Under port mappings put ***3100*** as Container Port as it is the default port of loki and either put specific port number as Host Port if you are using EC2 instances with Public IP and NOT utilizing ALB but we will be attaching it to ALB so we will set it to 0 which will set the host port dynamically ->
* 
* scroll down -> And check the Auto-configure CloudWatch Logs box so that ECS can automatically configure Log options -> Hit Update.
* 
* Scroll down and hit Create.

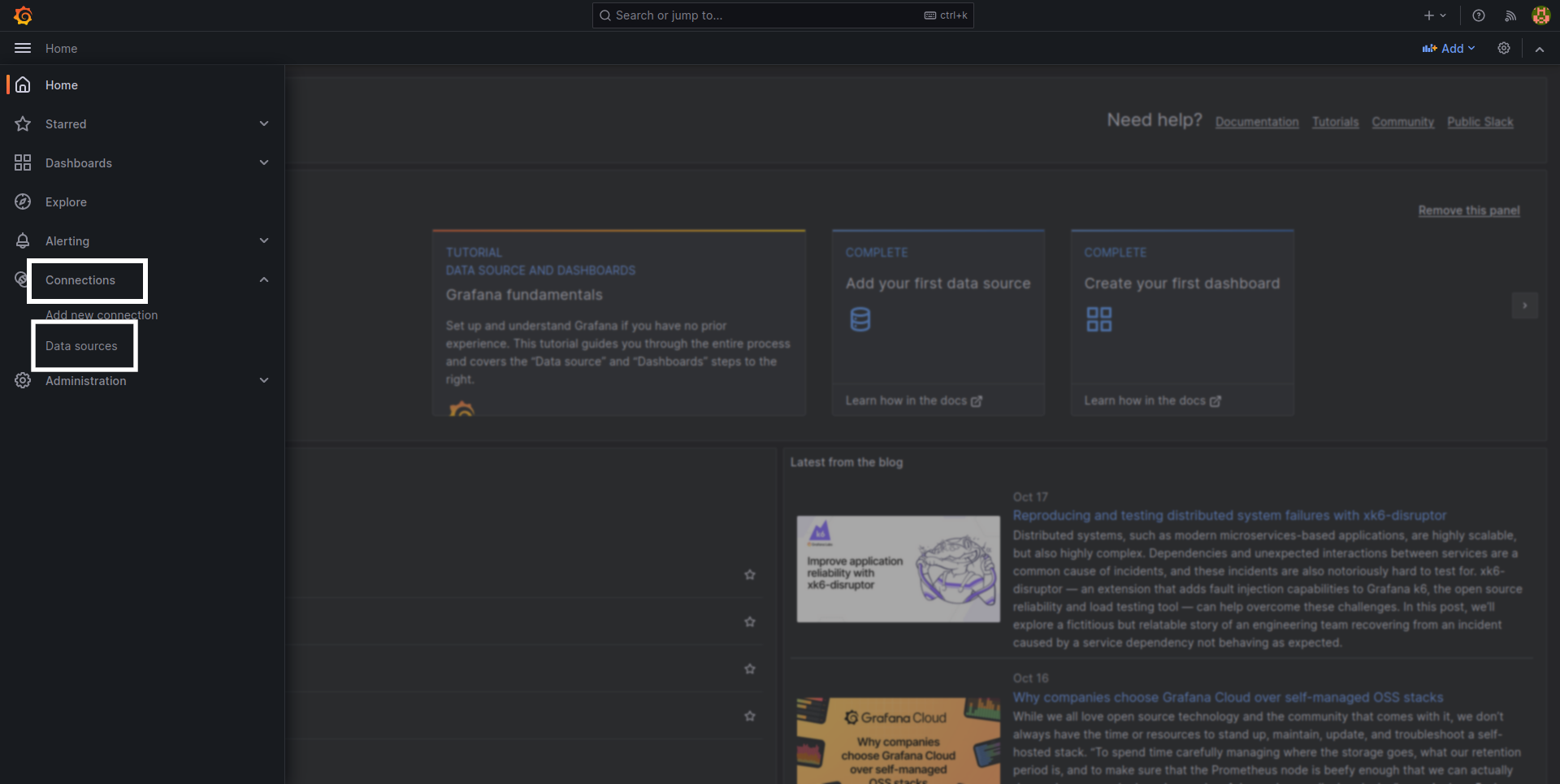
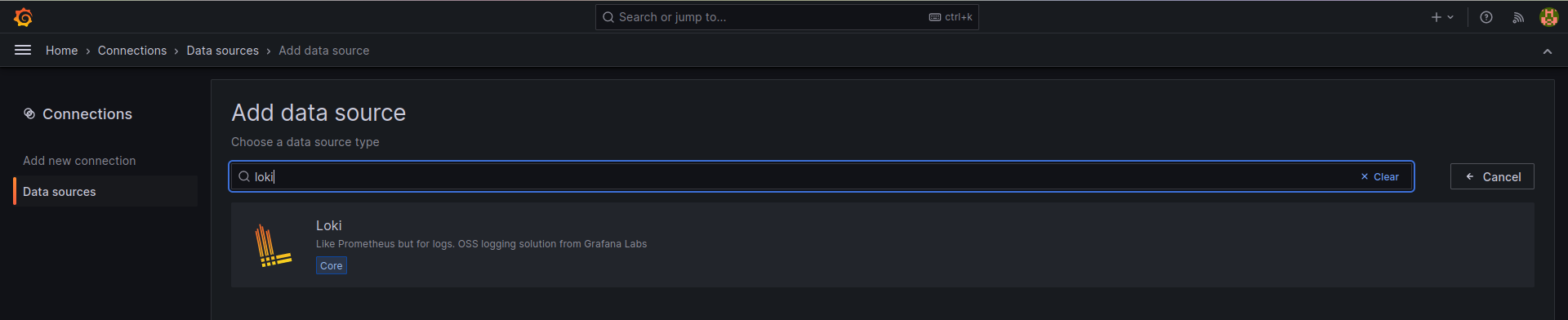
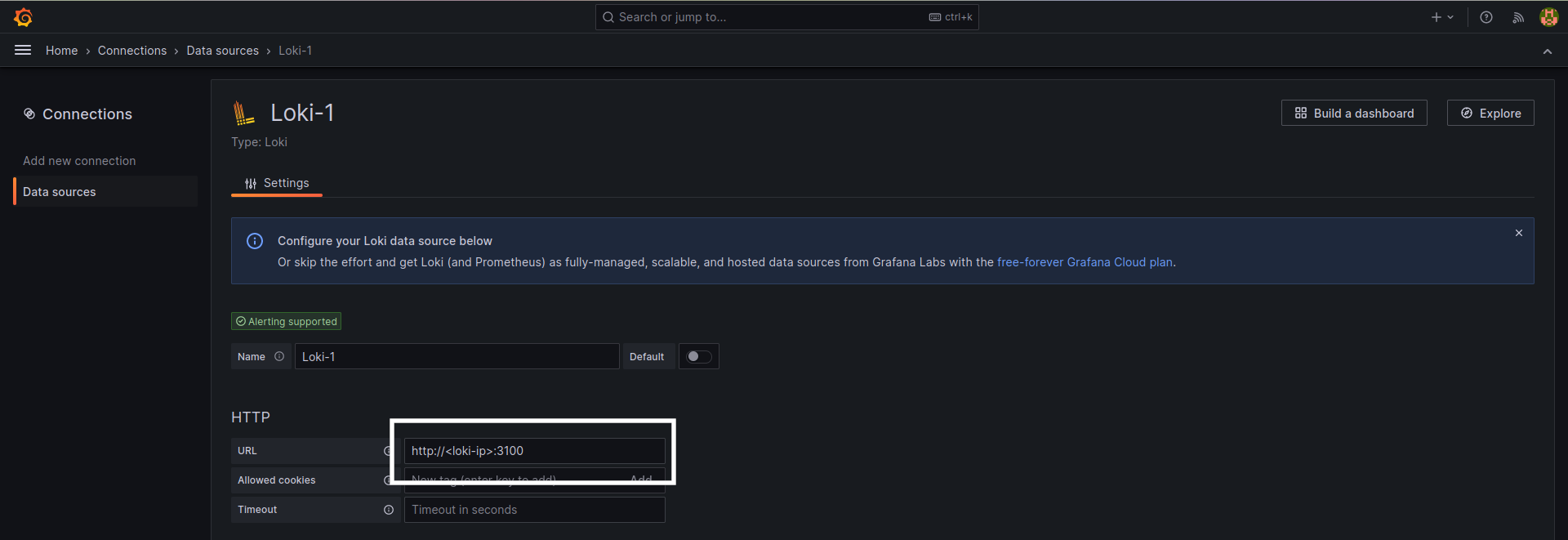
#### Deploy Loki ECS service.

* Refer Grafana Deployment step for this step as most of the steps are exactly the same. With only one minor change; that is in ***Health Check Path*** instead of */api/health* use ***/ready*** which is health check path of loki.

#### Modify ALB Rules.

* This step will be exactly similar as it was in the Grafana only change is the Host Header parameter which will be DNS of Loki and **NOT** of Grafana.

#### Integrate Loki With Grafana.

* Open Grafana in browser -> Navigate to Connections -> Select Data Sources.
* 
* Select “Add new data source” and Search for “Loki” -> Select Loki.
* 
* Add loki URL in the URL section. (**Note**\*: If your loki is using SSL use https instead of http and port, i.e. https://<loki>)
* 
* Hit ***Save & Test***.