**Speeding up with CloudFront and ElastiCache**

use aws services to speed up the per for app

**Cloudfront overview**

Steps to improve latency :

* Reduce distance between user and app
* global content delivery network designed to reduce latency.

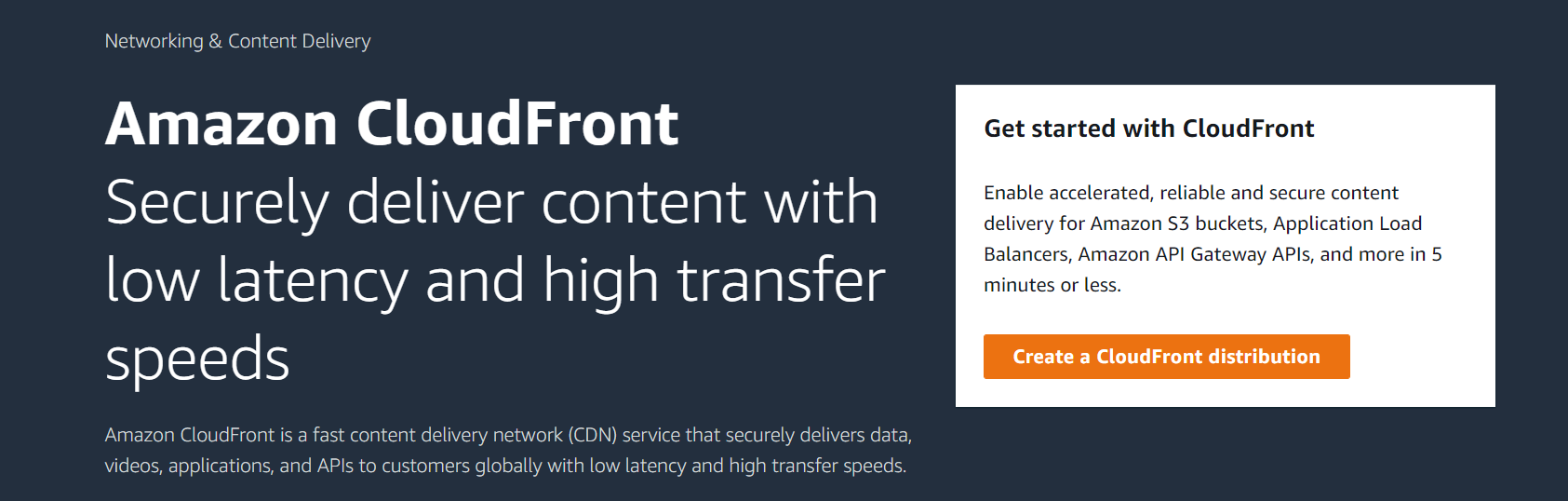
Cloud front - integrates with S3, EC2 and load balances. Edges "objects" and server them directly. proxies dynamic content to origin source

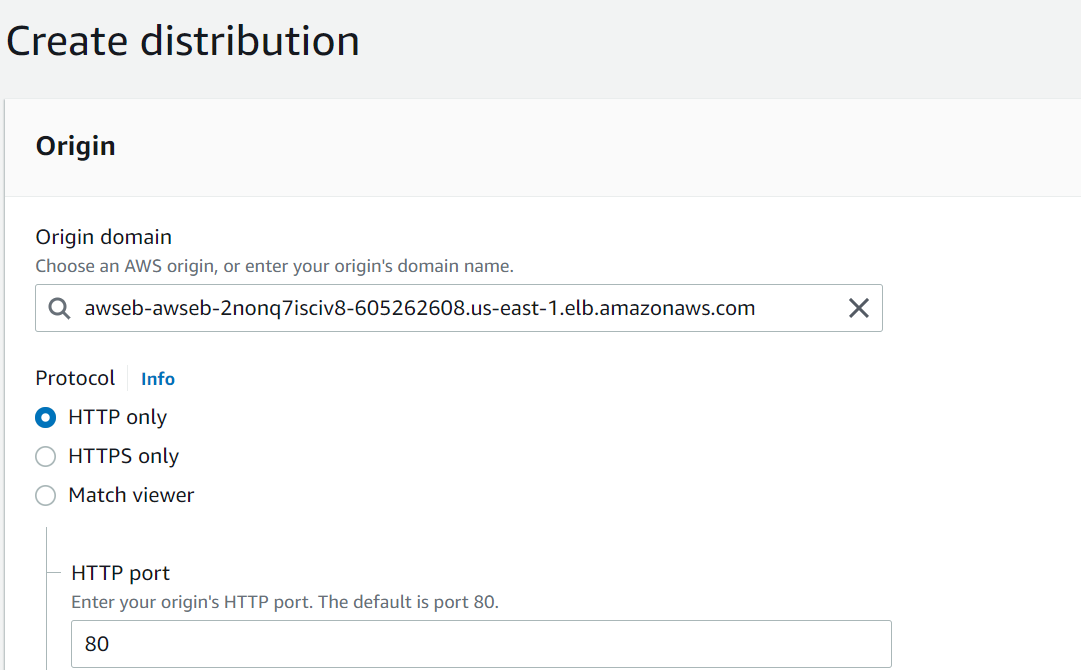
A set of content edged with cloudfront is called a distribution. a cloud front is given its own URL on the cloudfront.net domain.

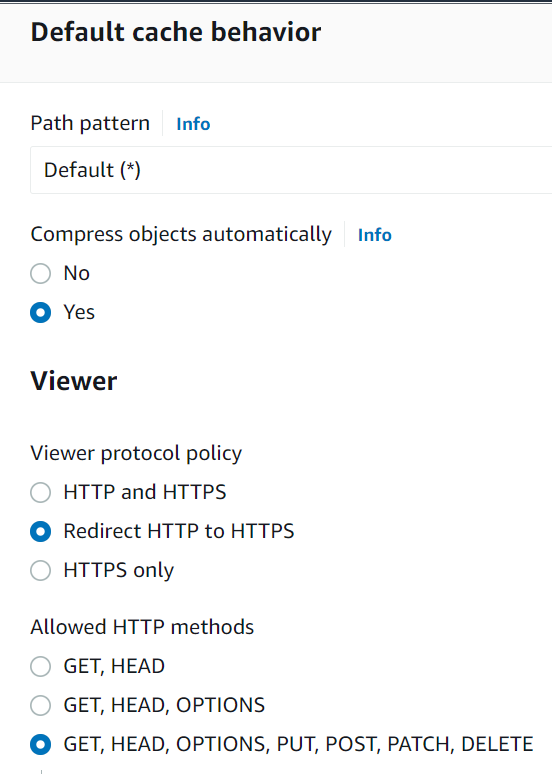
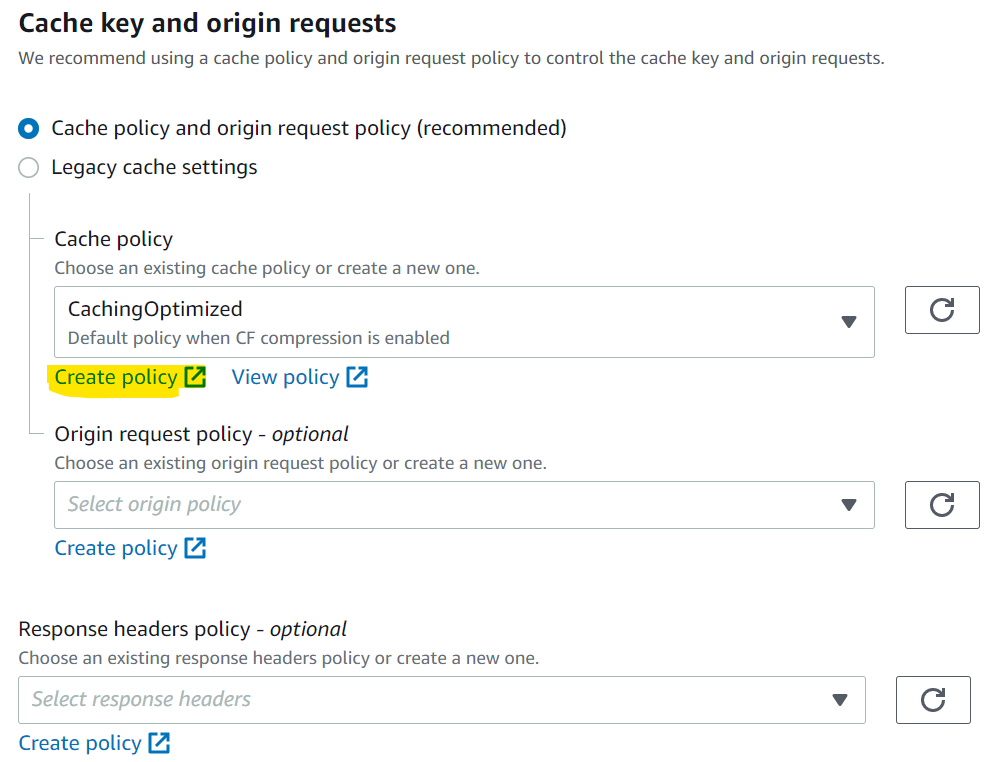
each distribution provides content from one or more origins like s3 buckets.

**Create the cloudfront distribution**

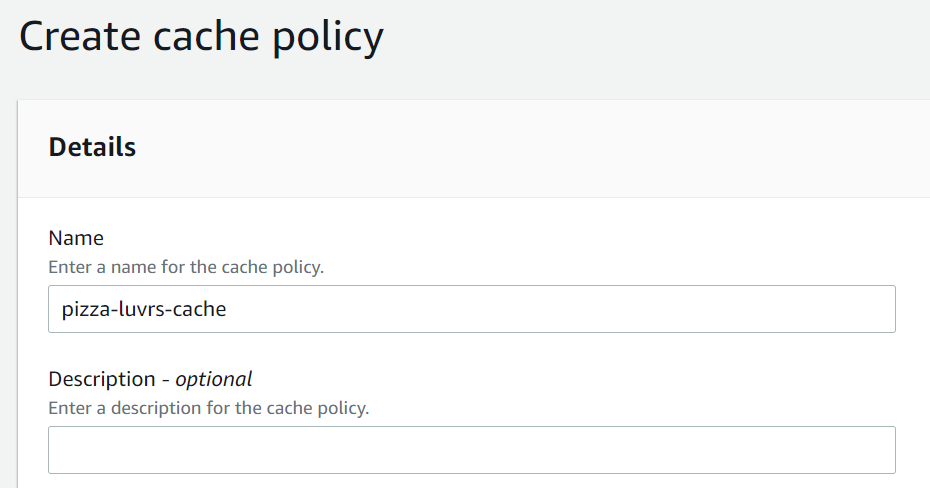
create a cloudfront dist to cache for the entire elastic beanstalk

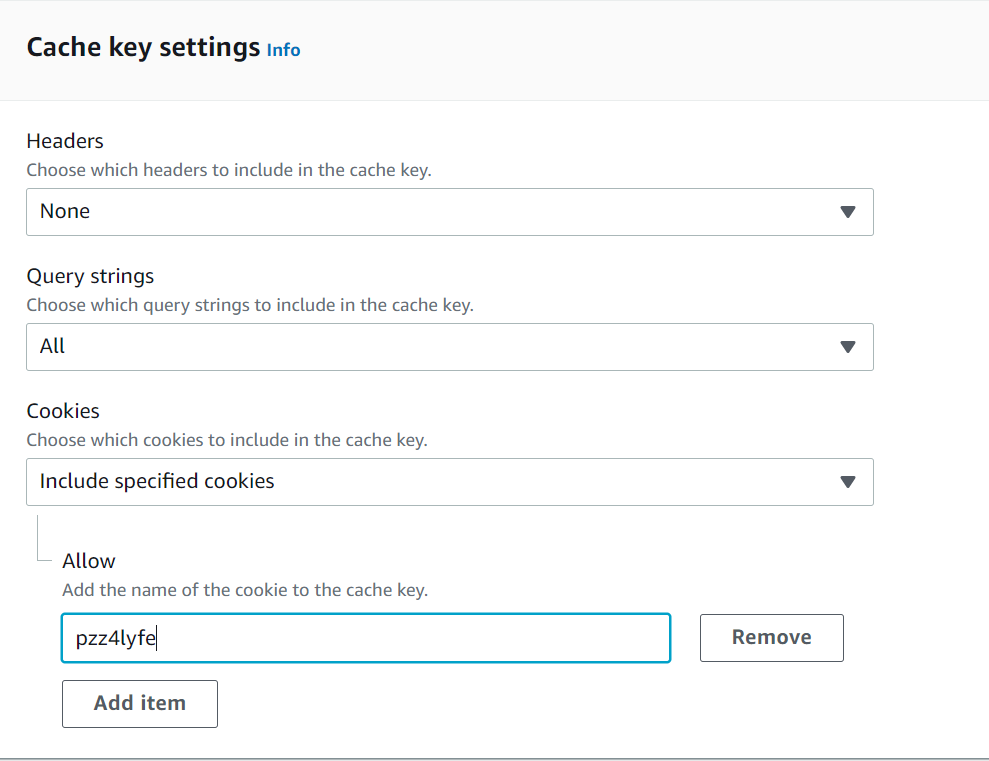


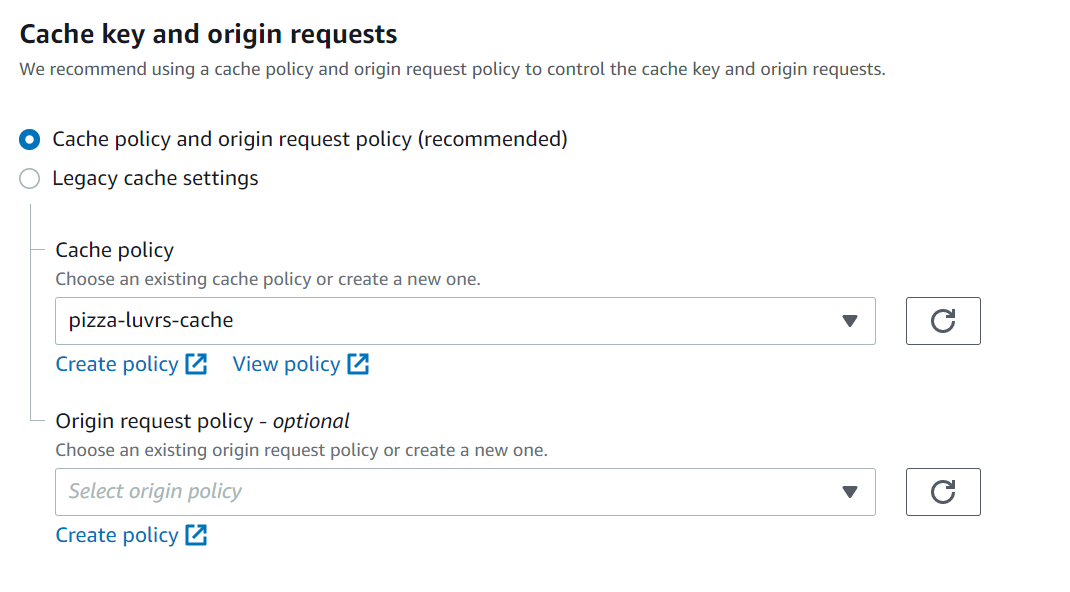


Cache behavior - define how cloudfront responds to req, caches content and handles cookies.







And create distribution

Origins tab list the origins that are providing content for the distribution but we can create more that one origin to be used with this single distribution.

**Elastic cache overview**

in memory cache

elastic cache - managed service for in memory cache datastore.

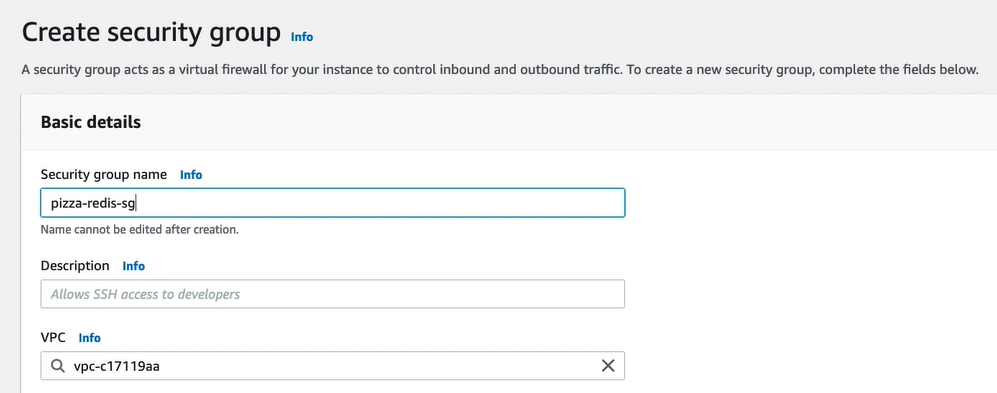
features : Managed maintenance, upgrades, etc. Automatic read replicas, simple node management.

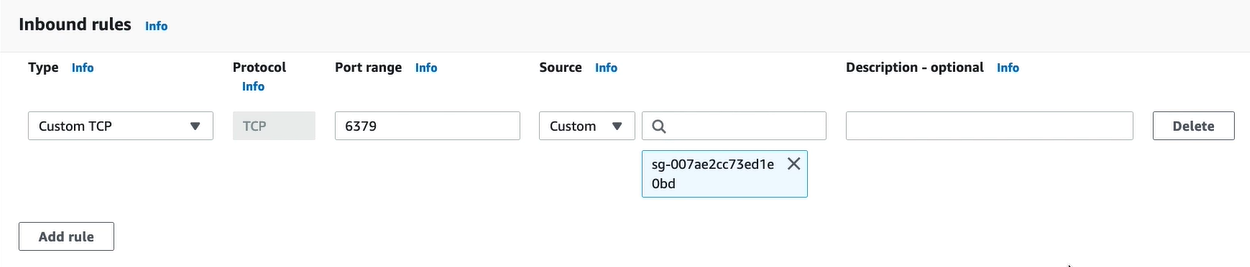
Elastic cache structure is a cluster. a cluster is a collection of nodes running a single cache instance

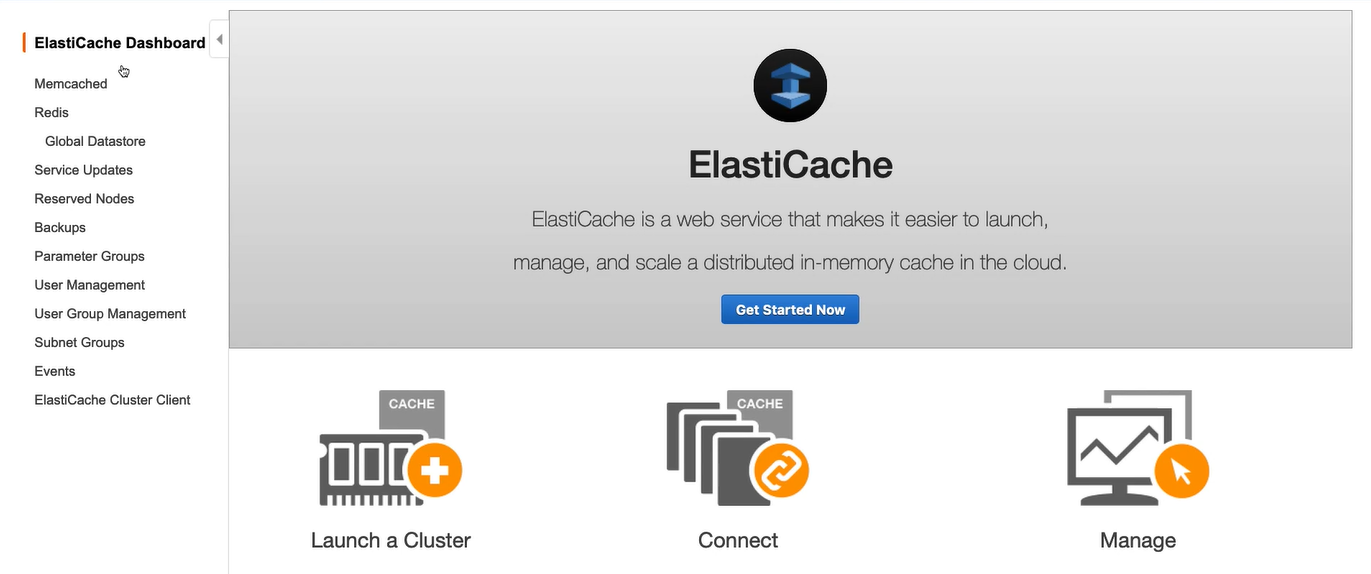
**Memcached cluser** comprise of 1-20 nodes nodes are basically individua EC2 instances running the caching software.

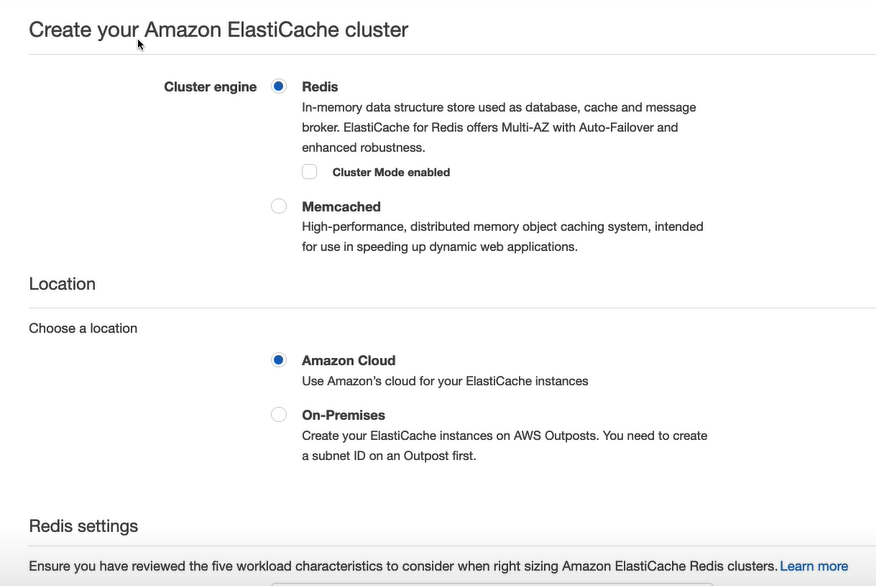
**Redis :** a cluster of one single node, and there are up to six read replicas tht can be also be added to create a replica group.

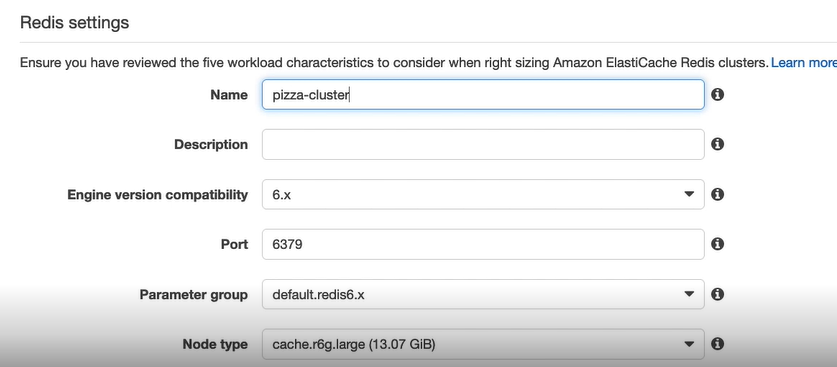
To create a redis first create a SG-security group

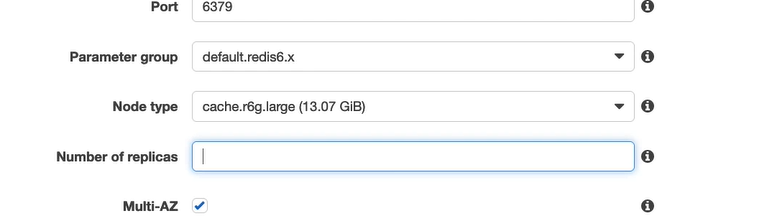




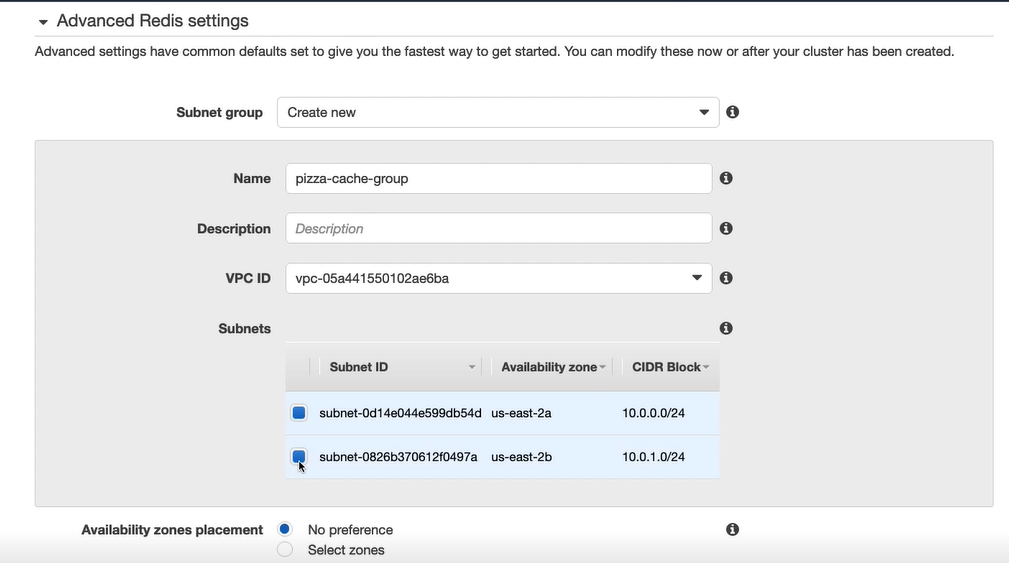


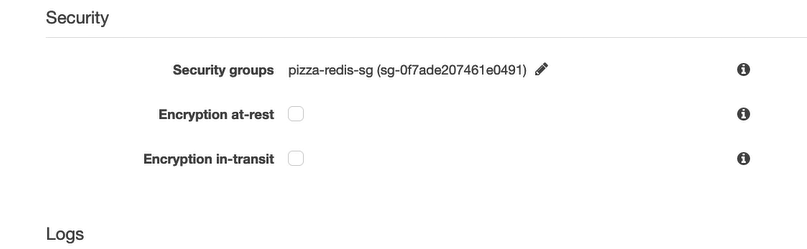






Select node type next





And create

