

روز دوم

رشته: پردازش ابری

مقدمه

ماژول: پیکربندی اولیه دیتابیس و in-memory datastore

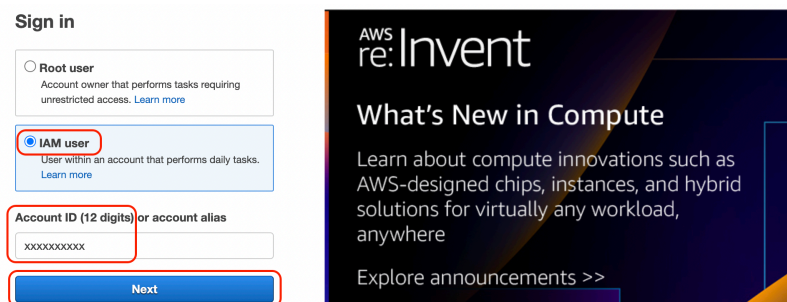
مدت زمان: ۴ ساعت و ۳۰ دقیقه

هدف این ماژول راه اندازی و پیکربندی اولیه دیتابیس ها و سرویس های Cache بر بستر زیرساخت ابری می باشد. در این پروژه شما می بایست نیازمندی های لازم برای ایجاد و کارکرد دو سرور ابری که در این سناریو نقش api server را شبیه سازی می کنند و اتصال آنها به mysql و redis را انجام دهید.

قوانین و استانداردها

- اطلاعات اولیه

- **اطلاعات ورود:** console.aws.amazon.com/console/home
- **نوع ورود:** IAM User
- **اکانت آی دی:** xxxx



- **نام کاربری IAM:** ()
- **رمز عبور:** ()
- **نام ریجن:** ()
- **شناسه رقابت کننده:** ()

- برای این مسابقه شما فقط و فقط باید از Region مخصوص خود استفاده کنید.

- اینترنتی که به شما اختصاص داده شده است فقط برای کارکرد و استفاده از پنل amazon aws می باشد و هرگونه

استفاده دیگر از آن (جستجو، بررسی داکيومنتیشن و ...) غیر مجاز می باشد.

- برخی از آیتم های سیستم نمره دهی کاملاً بر اساس ساختار اسمی مشخص شده تصحیح خواهند شد، پس دقت لازم را

به خرج دهید تا تمامی منابع ابری که در طی انجام این ماژول ایجاد می کنید هیچ کدام بی نام نباشند و نام آنها طبق

استاندارد و ساختار اسمی زیر باشد:

- [شماره]_[نام]_[نام سرویس آمازون]_[شناسه رقابت کننده]

▪ مثال: comp_20_ec2_myapp_1 ، comp_20_ksk_kafkasrv_2

سرویس Elastic Cloud Compute (EC2)

پنل زیرساخت ابری

داخل Region خود ۲ سرور ابری با مشخصات زیر را ایجاد کنید:

- نام سرور: apisrv (با در نظر گرفتن ساختار نام‌گذاری «قوانین و استانداردها»)
- هاست نیم در سیستم عامل: apisrv (با در نظر گرفتن ساختار نام‌گذاری «قوانین و استانداردها»)
- ایمج سیستم عامل: Linux Ubuntu 20.04 (free tier eligible)
- پکیج منابع: t3.micro
- فضای دیسک سیستم عامل: 8 GB
- نوع دیسک: General Purpose SSD (gp2)
- آی پی پابلیک: داشته باشد
- نحوه دسترسی به سرور: کلید خصوصی

داخل Region خود یک سرور ابری دیگر با مشخصات زیر را ایجاد کنید:

- نام سرور: internal (با در نظر گرفتن ساختار نام‌گذاری «قوانین و استانداردها»)
- هاست نیم در سیستم عامل: internal (با در نظر گرفتن ساختار نام‌گذاری «قوانین و استانداردها»)
- ایمج سیستم عامل: Linux Ubuntu 20.04 (free tier eligible)
- پکیج منابع: t3.micro
- فضای دیسک سیستم عامل: 8 GB
- نوع دیسک: General Purpose SSD (gp2)
- وضعیت شبکه: متصل به شبکه داخلی
- نحوه دسترسی به سرور: کلید خصوصی

یک آی پی شناور ایجاد کنید و آن را به سرور ابری internal متصل کنید. مطمئن شوید که امکان ssh به سرور از طریق این آی پی شناور وجود داشته باشد.

دسترسی عمومی برای تمامی سرویس‌های استفاده شده در این سناریو ایجاد شود.

- استفاده از Allow All Traffic و موارد مشابه مجاز نمی‌باشد.

شما باید طبق نیازمندی و سرویس‌های استفاده شده در پروژه، تنظیمات لازم را برای VPC و Security Group‌های خود برای شبکه داخلی و اینترنت انجام دهید.

سرویس Relational Database Service (RDS)

پنل زیرساخت ابری

یک دیتابیس با مشخصات زیر بسازید:

- نام دیتابیس: `mysqladb` (با در نظر گرفتن ساختار نام‌گذاری «قوانین و استانداردها»)
- نوع دیتابیس: `mysql`
- تمپلیت: `Dev/Test`
- نام کاربری ادمین دیتابیس: `admin`
- رمز عبور ادمین دیتابیس: شناسه رقابت‌کننده `Skills53`
- پکیج منابع و کلاس دیتابیس: `Standard – dbm6g.large/dbm5g.large – General Purpose SSD (gp2) – 20GB`
- وضعیت شبکه: از اینترنت قابل دسترسی باشد

بقیه تنظیمات مربوط به ساخت دیتابیس را می‌توانید با مقادیر پیش فرض تنظیم کنید.

پی‌کر بندی سرور ابری

سرور `apisrv_1`

- پکیج‌های لازم برای اتصال به دیتابیس را بر روی این سرور نصب کنید و به دیتابیس `mysqladb` که در RDS ایجاد کردید متصل شوید.
- یک دیتابیس با نام `dataset` ایجاد کنید.
- یک فایل با نام `dataset.sql` در اختیار شما قرار گرفته است که شامل اطلاعات برخی کاربران می‌باشد. این فایل را در دیتابیس `dataset` به طور کامل `import` کنید.

سرویس ElastiCache

پنل زیرساخت ابری

یک کلاستر `redis` با مشخصات زیر بسازید:

- نام کلاستر: `rediscache` (با در نظر گرفتن ساختار نام‌گذاری «قوانین و استانداردها»)
- تعداد عضو: ۳ عضو
- لوکیشن: `Amazon Cloud`
- پکیج منابع: `cache.r6g.large`

بقیه تنظیمات مربوط به ساخت کلاستر `redis` را می‌توانید با مقادیر پیش فرض تنظیم کنید.

سرور apisrv_2

- پکیج‌های لازم برای اتصال به redis را بر روی این سرور نصب کنید و به کلاستر redis cache که در ElastiCache ایجاد کردید متصل شوید:
 - o key/value زیر را در redis cache ثبت کنید:
 - skill53:index = "Today is the second day of Iran cloud computing competitions"
- یک وب‌سرویس با روش دلخواه (python, go, nginx, apache, ...) بر روی این سرور راه‌اندازی کنید.
 - o این وب‌سرویس باید هنگام serve صفحه پیش‌فرض، از redis مقدار skill53:index را فراخوانی کند و نمایش دهد.

مسیر کدی که فراخوانی از redis را انجام می‌دهد را در اینجا ذکر کنید:

تکمیلی سرور apisrv_1

- بر روی این سرور docker را با کمک اطلاعات زیر نصب کنید:
 - o **کلید GPG**
 - <https://download.docker.com/linux/ubuntu/gpg>
 - o **پارامترهای Repository**
 - `deb [arch=amd64 signed-by=/usr/share/keyrings/docker-archive-keyring.gpg] \`
`https://download.docker.com/linux/ubuntu focal stable`
- یک دایرکتوری با نام `/opt/container/production_application` ایجاد کنید.
 - o در این دایرکتوری شما می‌بایست تمامی متعلقات لازم برای ایجاد یک `dockerfile` و تولید یک `image` با مشخصات زیر استفاده کنید:
 - **مسیر داکر فایل:** `/opt/container/production_application/Dockerfile`
 - **توضیحات:** یک وب‌سرویس ساده با `golang` یا `python` بنویسید، بطوری که این وب‌سرویس در `http` `path` به مسیر `register` خود یک فرم داشته باشد و در این فرم فیلدهای استفاده شده از `tbl_users` دیتابیس را به کاربران نشان دهد. این صفحه از کاربران مقادیر این فیلدها را دریافت می‌کند و کاربران جدید را طبق ساختار این `table` در دیتابیس (RDS) ذخیره کند. (همچنین این امکان را دارید که در قالب `RESTful API` این کار را انجام دهید.)
 - همچنین داخل `/opt/container/production_application/` یک فایل `readme` کوچک با فرمت `markdown` ایجاد کنید و توضیحات لازم در خصوص صفحه ثبت‌نام یا `REST API` را به طور مختصر شرح دهید. در انتها شما باید یک `container` با نام `production_application` بصورت `running` بر روی این داشته باشید که از بیرون قابل دسترسی و استفاده باشد. همچنین تمامی `request log` های ورودی به این وب‌سرویس با فرمت زیر در `stdout` کانتینر نشان داده شود.
 - `timestamp - useragent - request_method - uri`