Oracle Autonomous Database technical overview.
overview of the Oracle Cloud Infrastructure
overview of the autonomous database.
key features of the Oracle Autonomous Database in the areas of self-driving, self-securing, and self-healing.
how the Oracle Autonomous Database integrates and fits with the Oracle Cloud Infrastructure.
put it all together with a typical workflow on deploying an autonomous database.

The Oracle Autonomous Database is the integration of the Oracle Database running on the Exadata platform with our complete infrastructure automation and our fully automated data center operations. Automated data sections include provisioning, patching, upgrading, and online backups, monitoring, scaling, diagnosing, performance tuning, optimizing, testing, and change management of complex applications and workloads, and automatically handling failures and errors.
Let's start with the Oracle Cloud Infrastructure. Because the Oracle Autonomous Database services are hosted as part of the Oracle Cloud Infrastructure, it is important to understand what OCI is, its key features, and how it integrates. So let's proceed with an overview of the OCI infrastructure.
Oracle built an enterprise cloud capable of running the most demanding and most innovative workloads. And we followed three key design principles. We knew that to be effective in supporting the system of records that run our customers' businesses, we need our infrastructure to be compatible with the critical and complex workloads our consumer base cares about as well as providing the same level of performance as what they have gotten on premises or better. That entails, first, industry-leading performance stats in terms of compute power and storage IOPS capability.
But even more importantly, in many ways, is the consistency of this performance. To effectively run stateful systems of record, performance can't be reduced by what's happening next to the customer. And it can't vary from the moment to moment, day to day, or month to month. To get this, we eliminate resource oversubscription from compute, memory, and network resources. This makes our cloud more expensive to build, but it gives our cloud the ability to run enterprise workloads more effectively than any other cloud today.
Oracle Cloud Infrastructure provides low, predictable pricing. We made the pricing of our cloud components low so that our customers could save money by moving to the cloud. But almost more importantly, we made the economics of our cloud far more predictable by making services all-inclusive and pushing autonomous services.
Oracle Cloud Infrastructure makes it easier to deploy Oracle products. Our existing customers can deploy faster and more easily and focus on using the product rather than the mundane tasks of managing and continuously upgrading the infrastructure. With tools and processes to help migrations, it makes it easier for Oracle customers to migrate to Oracle's cloud.

We built our cloud to support all the functionality and performance available in customer data centers, but with the benefits of increased agility, elimination of mundane tasks like managing hardware and facilities upgrades, patches, and capacity forecasting. We have deep expertise in cloud-specific automation. To make the migration possible without risk or high cost, we offer tools to connect our cloud to your data center to ours to enable the migration itself. Everything we run in our cloud is consistent with what you run in your own data center, including the Oracle Database itself, the surrounding ecosystem of tools like RAC, Data Guard, GoldenGate and all the third-party and management tools our customers use.

And we built it so that customers wouldn't take a step backwards in terms of performance when they move to the cloud. We give them the ability to run Exadata-engineered systems as a cloud service, offering the highest level of performance and scalability for Oracle workloads, something that is widely used on premises environments and not available in any other cloud. We will also build a cloud network with massive interconnect bandwidth and no resource oversubscription to ensure that noisy neighbors isn't an issue and high performance we deliver is invariable depending on external factors.
We provide a service-level objective that covers availability, performance, and manageability. We made it compatible with the key workload categories that our internal and external customers use. And there are four main workload categories our customers and partners run on our cloud. First is to move their implementations of Oracle applications to the cloud.
These are often complex, customized environments that can easily move to vanilla SaaS environments. We give these customers an easy path to move apps, as they run in their own data centers, to the cloud, where they get the same performance or better than on premises while no longer wasting time on hardware refreshes, system upgrades, or other mundane tasks. And they often save significant money as well. They get to bring all their customizations and easily integrate with other applications that also run in our cloud.
The next category is custom and ISV applications that run on the Oracle Database. There are thousands of enterprise organizations and software companies that use Oracle Database as a key foundation for applications they build. Oracle has made it easier for these organizations to build services that take advantage of our managed cloud database as well as the infrastructure optimized around this stack to reduce the level of effort they undertake in deploying these applications in our cloud. We can eliminate many of the mundane tasks of standing up and maintaining the database and the underlying hardware.
Our platform is also a great fit for performance- and data-intensive workloads. This includes true high-performance compute workloads of multiple varieties as well as data lake and other compute- and storage-intensive workloads where data access and consistent performance are key success criteria. Our design principle for cloud-native applications is to focus on leveraging the industry-

leading development streams in open source and elsewhere, making our cloud compatible with what customers are already using this with success.

With Oracle Kubernetes engine and registry for containers, customers can deploy the industry standard in container deployment and management on top of our predictable and performant bare metal infrastructure that avoids the conflict and performance degradation of hypervisors and server agents. We built an open-sourced-- our .fn project for serverless architectures, which can be downloaded and run anywhere, also available as a highly flexible and reliable cloud service. For management, we are heavily supporting Terraform from HashiCorp, a widely used infrastructure automation framework that can be used to program infrastructure deployments in our cloud as well as on premises and in other clouds.

Our approach is extremely comprehensive. Our cloud infrastructure provides all the core services to build and deploy production applications. Oracle has been building our PaaS services in our own infrastructure.

Services include compute, containers, storage, database, autonomous database, security, and integration. We also have an extensive SaaS offering, including CX, HCM, SCM, EPM, ERP, and data as a service. Since our hardware selection and design choices were focused on a dependable performance production applications need, it was easy for us to also cover performance-intensive workloads, including any HPC workload, even those requiring specialized hardware. And if you're building new cloud-native applications utilizing functions, Docker, or Kubernetes, we have those services as well.

Oracle Cloud Infrastructure is hosted in regions and availability domains. A region is a localized geographic area. And an availability domain is one or more data centers located within a region. Our region is composed of one or more available domains. Most cloud infrastructure resources are either region-specific, such as virtual cloud networks, or availability-domain-specific, such as compute instance.

Traffic between availability domains and between regions is encrypted. Availability domains are isolated from each other, fault tolerant, and very unlikely to fail simultaneously. Because availability domains do not share infrastructure such as power or cooling, or the internal availability domain network, a failure at one domain within a region is unlikely to impact the availability of the others within the same region. The availability domains within the same region are connected to each other by a low-latency, high-bandwidth network, which makes it possible for you to provide high-availability connectivity to the internet and on premises and to build replicated systems in multiple availability domains for both high availability and disaster recovery. Regions are completely independent of other regions and can be separated by vast distances across countries or even continents.

A fault domain is a grouping of hardware and infrastructure within an availability domain. Each availability domain contains three fault domains. Fault domains let you distribute your instances so that they are not on the same physical hardware within a single availability domain. A hardware failure or compute hardware maintenance that affects one domain does not affect instances on other fault domains.

To control the placement of your compute bare metal DB system or virtual machine DB system instances, you can optionally specify the fault domain for a new instance at launch time. If you do not specify the fault domain, the system selects one for you. To change the fault domain for an instance, terminate it and launch a new instance in the preferred fault domain. Use fault domains to, number one, protect against unexpected hardware failures, number two, protect against plant outages due to compute hardware maintenance.

Oracle offers a broad variety of compute solutions from small and virtualized to very large and dedicated, from web servers to high-performance application servers, with either network block storage or local non-volatile memory. These options enable you to build a range of applications on the same high-performance network from traditional enterprise to modern scale-out, from unpredictable to steady state. Virtual machines and bare metal compute with predictable IOPS, block storage for general purpose needs-- these standard options include new instances based on AMD EPYC processors, which cost less than half of our other VM offering, and higher bare metal core counts.

Dense IOB virtual machines with local non-volatile memory storage provide a range of compute and capacities with high IOPS, bare metal GPUs with two P100 and eight P100 GPUs, 28 to 52 cores, virtual memory GPU options, and predictable IOPS block storage, bare metal compute with 52 cores, high memory, and optional local non-modeled SSD provisioning in under five minutes. And there's a specialty HPC SKU with higher all-core Turbo Core frequencies and RDBMA capabilities.

Oracle Cloud Infrastructure also provides a production-ready RDBMA network in the cloud, enabling us to serve tightly coupled HPC workloads as well as easily parallelizable ones. The cluster network is an RDBMA-based network that lets you form clusters of compute, storage, GPU, or hybrid that use secure, ultra-low latency networks between cluster nodes. This allows complex CFD or simulation workloads to run on OCI targeted for the hardest product development workloads such as CFD, crash simulations, reservoir modeling, or DNA sequencing.

Oracle Cloud provides optimized storage for nearly any use case. Local, non-volatile SSD provides the fastest performance for transactional database and HPC use cases. File storage offers a managed file storage service that scales from just kilobytes of data to exabytes, making it ideal for enterprise applications, big data, analytics, scale in applications, and container-based applications. Block storage is the most flexible for application development and deployment in classic tiered applications. Object storage provides great economics for backup and archive as well as big data lakes. Whether your application prefers a tiered storage strategy with snapshots, backups, and replications, or more of a scale-up model, OCI offers a wide range of highly performant options.

Oracle Cloud Services make security a top priority. Security is broken down into four areas- - number one, deeper customer isolation to prevent customer peering or accidental data sharing; number two, data encryption end to end, data not be viewed by non-authorized users; number three, network protection to prevent access to applications and data; number four, verifiable security for full accountability of access to any resource to comply with regulations and for forensic analysis.

In this light, you can see a brief overview of our extensive services. This is the true enterprise cloud you've heard mentioned. We don't focus on micro-instances or VMs with time sliced, fractional CPU allocations. We focus on providing what business needs to run real production workloads, workflows that have to scale up as well as out, workloads that may require the reliability of a solid, traditional hardware infrastructure in addition to the plentiful approach of cloud, workloads that need low-latency access to storage and networks. And we provide businesses with simple pricing and predictable costs instead of an arcane system that penalizes you for running the high-performance production applications you depend on.

And once you get to our cloud, then the innovations kick into high gear. Customers have a full range of options to deprecate and eliminate their data centers if they choose to or to keep them running for some workloads with deep compatibility and connectivity options with the Oracle Cloud. We allow customers to expand their curation of data with deep analytics and integration options to get into Oracle's new Autonomous Database Cloud service that eliminates tedious management tasks and represents the future of enterprise data management.

Customers can augment their own data with data we own in our Oracle Data as a Service Cloud. Customers can also expand their network of applications, surrounding their data with cloud-native functionality that allows them to build new, innovative approaches to managing and making use of data, including our Kubernetes-based container service, our flexible .fn serverless capabilities, as well as a broad ecosystem of third-party options that unlock new value from data.

Let's look at the Autonomous Database. Transform from building and maintaining database to using autonomous services and modern clouds. This allows you to, one, innovate faster with lower costs, developing and optimizing new application faster, cutting runtime costs up to 90%, eliminate full-stack administration costs, and number two, ensure data safety by eliminating cyber-attack vulnerabilities and obataining service-level objectives of 99.95%.

Starting in Oracle Database 9i, we began to introduce, and matured, many sophisticated automation capabilities from memory management to workload monitoring and tuning, all of which are used in the Autonomous Database. But it's not just the database management that Oracle has been automating. We have also spent the last decade working on the database infrastructure with our engineered systems, which provide the best platform for the Oracle Database as they are only preconfigured, pretested, and optimized platforms for the database.

Oracle Autonomous Database is actually a family of cloud services with each member of the family optimized by workload. The first member of the family is the Autonomous Data Warehouse, which has been optimized for analytic workloads such as data warehouse, data marts, or as part of a data lake. The second member is the Autonomous Transaction Processing. ATP is optimized for transaction processing or mixed workload environments and makes an excellent platform for new application development.

All members of the Autonomous Database family share the same fully automated, high-performance Exadata infrastructure that provides world-class availability and scalability. They also share complete automation of all database administration tasks, such as provisioning, patching, securing, backups, et cetera. Autonomous Transaction Processing only difference from Autonomous Database Warehouse when it comes to how to optimize for each specific workload within the database.

Autonomous Data Warehouse is optimized for complex analytics, while Autonomoous Transaction Processing is optimized for high-throughput transaction processing. When you start loading data into the Autonomous Database, we store data in the appropriate format for each workload. If it's in Autonomous Data Warehouse, then we store data in columnar format. And that's the best format for analytics processing. If it's Autonomous Transaction Processing, then we store the data in row format as that's the best format for single-row lookups.

In terms of data access, in Autonomous Data Warehouse, we use data summaries like storage indexes on the Exadata storage cells. And the result cache quickly accesses only the data needed to answer each query. On Autonomous Transaction Processing, the indexes are used to access only the rows, or records, needed for each transaction. For data processing, analytic workloads, we automatically parallelize the query execution to access large volumes of data in a short amount of time to answer business questions. If it's transaction processing, then we'll automatically use indexes to quick access the appropriate records. We will also detect missing indexes and create them for you.

In terms of memory, Autonomous Data Warehouse uses the data set large to cache. So memory is used to speed up large joins and aggregations such as group-by operations. On Autonomous Transaction Processing, we use a majority of the memory to cache the active data set to av

oid any IO. We also use RDBMA to access data directly in me mory on the other service in the RAC cluster.
Regardless of the workload, we need to keep optimizer statist ics current to ensure we get optimal execution plans. With AD W, we're able to achieve this by gathering statistics as part of the bulk load activities. With ATP, where data is added using more traditional insert statements, statistics are gathered aut omatically periodically. As the data volume changes, or new a ccess structures are created, there is the potential for executi on plans to change. And any change could result in performan ce regression. So we use Oracle SQL plan management to ens ure that plans only change for the better.
Now that you have learned how autonomous came to be, let's take a deepe r look at the key features. Oracle Autonomous Database provi des many benefits for business users and administrators, star ting with the lower operation costs due to optimized and on-demand sizing configurations that only get billed when used b y the hour. It provides entry-level configurations of just one CPU and 1 terabyte of allocate d database space, and includes backups, analytics, and develo pment tools, and full management for the service price. Ther e is substantial risk reduction by running an Oracle Autonomo us Database due to the several risk mitigation strategies inclu ded with the service.
All data is encrypted at rest and in communication. Defined u ser roles ensure no accidental data inspections by non-authorized users. Application of security patches as soon as t hey are available and robust security around the Oracle Cloud operations combine to reduce many areas of risks normally a ssociated with on-premises installation. The very fast provisioning and availabili ty of an Oracle Autonomous Database and its ease of access f rom anywhere with internet connectivity makes it an excellen t platform for innovation, much faster than installing and impl ementing an Oracle database on premises. We will discuss thi s in more detail in another module.
Instantiating an Oracle Database on the Oracle Cloud and onl y takes a few steps and minutes, making it simple to impleme nt. If you have an existing database application, it can be expo rted and imported into autonomous database through a fully automated process in the included SQL developer tool, allowi ng you to re-point application servers to the autonomous database and ha ve a quick migration to the cloud. Autonomous database runs on Oracle's highly optimized database infrastructure, Exadata , which makes it the fastest Oracle Database platform in the cl oud while providing on-demand elasticity by allowing customers to add both computi ng and storage as needed, when needed, without sustaining a n outage. This topic will be further covered in detail in a futur e module.
The mission of the autonomous database is to provide a servi ce that is self-driving, which will automatically take care of all database and infrastructure management as well as monitoring and tuning. So the user will simply specify the service-level agreement, and Oracle will make it happen. We believe this will help reduce costs and improve productivity by autom ating the mundane tasks of having to provision, patch, and ba ck
up databases. Freeing up their IT teams to focus on the task w ill bring value to the business.
We also want the database to be self-securing, protecting itself from both external and internal mal icious attacks. We do this by automating encryption of all dat a, whether it's at rest or in flight and automatically applying s ecurity updates with no downtime. Finally, we want the auto nomous database to be self-repairing. And by that, we mean it will automatically recover from any failure and minimize all kinds of downtime, including planned maintenance, with an S LA guarantee of 99.95% availability. That's less than 30 minut es downtime per year, including planned maintenance. It will also elastically scale compute or storage.
The autonomous database self-driving capabilities include rapid provision, self-scaling, automatic tuning, and automatic indexing. Together, t hese capabilities provide the ability to provision a database in minutes with automated management, monitoring, and tuni ng. It provides the time for you to focus on innovation instead of daily mundane tasks. Let's take a look at the capabilities in greater detail.
It is to provision a new industry great proven database that us es RAC and Exadata in minutes. Oracle applies all the best pra ctices of 40 years in this database. You don't have to worry ab out configuring, applying, tuning, installing the hardware, or s oftware, or anything. It's all taken care of for you. You select CPU and storage separately-
- so scaling independent when you need more CPU resources or just additional storage.

In autonomous database, optimizer statistics are gathered au tomatically during direct-path load operations. If users need additional statistics, they c an gather stats manually at any time. Machine learning also al lows autonomous database to optimize executions based on usage patterns of each database. Because autonomous datab ase and the Exadata platform it runs on are so efficient at run ning the Oracle Database, by default, optimizer and parallel hi nts are ignored. Parallelism generally is determined by define d services in the autonomous database-
- more of this in future modules.
Users have the ability to explicitly re-enable hint processing if it is required for specific reasons. Alt hough autonomous database is designed to completely auto mate and provide the best environment for running the Oracl e database applications, Oracle realizes there may be specific reasons, such as application compatibility or referential integr ity, where items such as indexes may be required. So let's revi ew where Oracle recommends self-tuning services provided by the autonomous database.
Number one, tables do not need to be partitioned. And partiti oning should not be used as a performance-enhancing design objective in autonomous database deploym ents. Databases that are being migrated to autonomous data

base should have partitioning removed unless there's a specif ic operation reason for use.
Number two, in general, indexes should not be used on tables for performance reasons. Autonomous database and the Exa data platform it runs on provide automatic enhanced indexin g for data retrieval that, in most cases, performs better than manual indexing. Number three, autonomous databases use compression of data in the database. So additional compressi on does not need to be used.
Number four, in-
memory tables cannot be used in autonomous database. And number five, tables spaces do not need to be created. Manua l tuning of partitioning, indexes, materialized views, and com pression is available, but should only be used with careful con sideration, such as in cases where migration of an existing sys tem whose data loading scripts rely on partitioning or indexin g is used for referential integrity.
Oracle execution plans are like driving directions. They will ch ange as the data distribution changes-
- data volumes and statistics. Indexes can be thought of as ro ads and bridges. With auto-
indexing, new roads will be added as the workload continues. Changes in data volume and SQL workloads are continuously captured. And machine learning algorithm processes changes to find new optimal plans and indexes.
An expert system that implements indexes based on what a s killed performance engineer would do is part of the environm ent. It first captures, periodically, the application's SQL history into a SQL repository and includes SQL plans, bind values, ex ecution statistics, et cetera. It then identifies candidates for in dexes that may benefit the newly-
captured SQL statements. It creates the index candidates as u sable and invisible indexes, metadata only. And it drops index es obsoleted. by the newly created indexes, performing a logi cal merge.
Third step is to verify these new indexes. At this point, yes, th e optimizer index candidates will be used for captured SQL st atements, materialize indexes, and run SQL to validate that th e indexes improve the performance. And all verification is do ne outside application workflow.
At that point, there's a decision to be made. If the performan ce is better for all statements, the indexes are marked visible. If performance is worse for all statements, the indexes remai n invisible. If performances were for some, the indexes are m arked visible except for the SQL statements that regressed.
There is a monitor capability which monitors index usage in c ontinuous mode, automatically creates-
- the indexes that have not been used in a long time will be dr opped. You can switch this service off. It is resource-
controlled, so it will only use one CPU for doing auto-
indexing. If we take a copy of our data and run on our CPUs, t hen we may run into security and data privacy issues.

Automatic indexing creates secondary indexes that are used t o improve SQL performance other than primary key and forei gn key indexes. It applies to tuned and untuned applications. For tuned applications, existing secondary indexes may be ou tdated, or an important one can be missing. Some secondary indexes m ay also be dropped if they are no longer useful.
For untuned applications, development frameworks and obje ct relational mappers often only generate primary key indexe s, and sometimes, foreign key indexes. Auto-
indexing augments existing primary and/or foreign key constr aints to improve performance. It supports single-
column and concatenated indexes, function-
based indexes, and compression advanced load. In this exam ple, you can see how ATP created 43 auto-
indexes in just 30 minutes to improve performance.
Two functions of the automated management feature of ADB are backups and patching. Backups are scheduled on a nightl y basis to Database Backup Cloud Service with a retention per iod of 60 days. The cost of backup and storage is included wit h the price of ADB. The GUI console shows detailed informati on about backups that have been taken and allows restores fr om any of them. Full-
stack patching is done once a quarter in rolling fashion across nodes of cluster to maintain the availability of the service. Ti me to apply patches is automatically selected by Oracle Cloud operations, but customers can override the selection and sel ect an alternate time.
Let's move to self-
securing and dive into what helps the database to be self-
securing to protect all your data. This section highlights the b enefits of self-
securing and the key Oracle technologies and capabilities that enable them. This is certainly not an exhaustive list. But thes e are the key capabilities that we will be diving into in this sec tion and understanding how they work to make the database self-securing.
Autonomous database stores all data in encrypted format in t he Oracle Database. Only authenticated users and application s can access the data when they connect to the database. All connections to the autonomous database use certificate-
based authentication and Secure Socket Layer, SSL. This ensur es that there is no unauthorized access to the autonomous da tabase and that communications between the client and serv er are fully encrypted and cannot be intercepted or altered. C ertificate-
based authentication uses an encrypted key stored in a wallet on both the client, where the application is running, and the server, where your database service on the autonomous data base is running.
The key on the client must match the key on the server to ma ke a connection. A wallet contains a collection of files, includi ng the key and other information, needed to connect to your database service in autonomous database. For data encryptio n security keys, Oracle allows separation of keys. For encrypti on at rest, Oracle allows you to enable and disable encryption . Oracle delivers it, by default, in the on option.
Patching is very expensive, because it requires downtime and several man hours to patch all the databases in your environ ment. Also, patches may be applied once in a quarter. So it's an ongoing effort. Autonomous database will patch your syst ems for you while the database is running. It needs no downti me and manual effort. So it can never happen that you forget

to apply the patch or didn't have time to do so. It ensures that you're always protected from known cyber-attacks.
In autonomous database, no logins are allowed to the OS. No root or SYSDBA logins are allowed. The only allowed logins are as admin, privileged default autonomous database user, or regular database user. No call-outs to the OS are allowed from autonomous database. This prevents installing or modifying any software on the system. Database clients can connect securely using a TLS wallet. Databases in dedicated autonomous database run in customer private virtual cloud networks to prevent network access by other customers or hackers.
Public IP is not required. Secure configurations are deployed at all levels of autonomous database- - OS, database, storage, et cetera. Oracle automatically applies updates and the latest security patches on a quarterly or off cycle for high-impact security vulnerability. Native encryption prevents data access from outside the database.
Oracle tools leveraged by autonomous database for security are Data Masking and Database Vault, which accomplish, first of all, no access to the database node or file system- - Oracle DBAs are separated from actual data- - number two, dynamic reduction and masking of data- - Oracle can apply security policies as data leaves the database- - for example, convert social security number to a representation like xx and the last four digits of the number. Number three, static masking for test dev databases can simply convert sensitive fields. Number four, metadata tagging- - this is part of the label security option. Data can be marked as sensitive, confidential, et cetera. And it is included for free in the Database Cloud Service.
And number five, full defense in-depth, it is built over 30 years of meeting the needs of the most demanding organizations, high-threat environments, security services, and financial institutions. Database auditing is configured by default and customizable to meet your needs. Autonomous database comes preconfigured using Oracle Unified Audit. This feature includes automated auditing for privileged user activity and login failures and optional preconfigured policies for the Center for Internet Security audit benchmarks, account management, and much more.
The audit trail is available through service REST call invocations. Database audit trails can also be retrieved. Future release will include detailed auditing through additional security services. The autonomous database provides preventive protection against all unplanned and planned downtime and rapid, automatic recovery from outages without downtime.
There is a broad range of events that can cause database downtime, including component, storage, and servers failures, database crashes, or even site-wide or regional outages due to a natural or man-made disaster, data corruption that can cause incomplete backups or render the data useless, human error, which plays a significant role in many cases, whether it's a database table that was dropped, a cable that was accidentally unplugged, or a tape that was lost, planned downtime for patching, upgrades and maintenance, which represents an increasingly disproportionate percentage of overall downtime for many growing organizations.
Oracle has successfully addressed all of these causes of downtime and disruption in on-premises environments for decades with the Oracle Maximum Availability Architectural, Oracle MAA. Oracle MAA is a set of advanced technologies and best practices that can be deployed to handle any service-level requirement, with solutions ranging from periodic backups to zero data loss and zero-downtime-replication-based disaster recovery. The MAA portfolio is also available in the Oracle Cloud and has been enhanced with automated functionality that minimizes-- in many cases, eliminates- - human intervention.
Exadata not only continuously monitors for failing devices, it also provides redundant database servers that provide active, highly available cluster servers, hot-swappable power supplies and fans, redundant power distribution units, provides redundant storage grids that provide data mirrored across storage servers, and redundant, non-blocking IO paths, and redundant networks that include redundant IB connections and switches. The self-healing software automatically runs all monitoring and fault prevention tools in the background 24 hours a day, seven days a week.
It uses Oracle's 40 years of experience to build machine learning models to make sure they monitor the system and make sure the system is providing the maximum availability and healing. it applies machine learning algorithms and Oracle's best practices to fully automate database operations. Oracle uses machine learning algorithms like anomaly detection, pattern recognition, problem signatures to detect and prevent issues and failures and fix the known issues or erase SRs of the half of our customers. Bug detection should be our job, not yours.
It can correlate problems across different systems to give you a complete story of the fault that occurs. The hardware supports itself and heals itself. With cloud-based, region-based duplication everywhere, database hardening, RAC redundant compute, triple-mirrored storage, we can provide 99.95% service level objective through the stack. All these technologies that make the Oracle Database highly available are now provided to you with the autonomous database. So you don't have to think about network failures, hardware failures, failing disks, or if your host fails, on taking backups, or even if your entire region sinks in an earthquake.

We have now reviewed the Oracle Cloud Infrastructure and the autonomous database key features. Now let's take a deeper look at the architectural component. The autonomous database is placed on an Exadata system based on the region where the customer is located or closest to. This placement, except for region location, is invisible to the customer, but is done to minimize traffic latency and maximize data center efficiency.

Oracle completely manages and controls all operation aspects of the system, including patching, software versions, isolation, backups, and other operational procedures. This provides the most flexible and least obtrusive environment for the customer and the most effective environment for Oracle Cloud Operations to manage. This also allows Oracle to offer this service with a very minimum required description of one OCPU and 1 terabyte of storage, which, while already providing a sizable development environment, is a very low cost of entry. The minimum commitment for a customer to the environment is one hour of built usage time.

As previously discussed, Oracle Autonomous Database runs on Exadata systems hosted on Oracle Cloud Infrastructure data centers. Oracle Autonomous Database storage is on Exadata Storage Servers, which are directly attached to the Exadata compute nodes. A complete autonomous ecosystem also consists of dedicated OCI servers that run the Oracle Machine Learning environments. Oracle Machine Learning environments can be accessed from the autonomous database cloud console or directly through the URL provided when an [INAUDIBLE] user gets provisioned.

When a user or other process connects to the autonomous database, the connections are routed through connection manager servers that distribute and manage connections into the databases in the Exadata servers. These connection managers are attached and connected to the network infrastructure. Oracle Cloud Infrastructure physical or virtual servers that run applications that leverage databases in autonomous database are also a typical, but not required, component of a fully integrated autonomous database cloud stack.

Although the autonomous database environment is hosted on Exadata systems, which, themselves, provide high availability via the nature of the Exadata architecture, the Oracle Cloud Infrastructure data centers on which they are hosted provide an additional level of availability through its availability domains. Autonomous database is hosted in regions and availability domains. A region is a localized geographic area, and an availability domain is one or more data centers located within a region.

Each autonomous database environment comprises of several components, including the Exadata servers, Exadata Storage Servers, Oracle Machine Learning Servers, the CMAN and shared servers. This same architecture is replicated across availability domains to provide services redundancy. Users of the autonomous database are matched their service through load balancers that distribute load across available services to provide equal distribution of these available resources.

Connectivity to object storage such as Amazon S3 also leverage this architecture. Connectivity to the autonomous database for permitted access, such as those of Cloud Oracle Operations, virtual cloud networks, or shared services are performed through a whitelisted IP service that guarantees that only determined IP addresses can access these services directly.

Supporting storage for databases on the autonomous database are provided by the Oracle Cloud Infrastructure Object Storage service. Oracle Autonomous Database performs automatic backups on provisioned databases. And those backups get stored in private storage defined in the Oracle Object Store. Backups are automatic and non-optional an autonomous database. And no setup is required by users.

However, autonomous database allows users to create their own additional backups for other operational purposes, including point in time recover if needed. And that backup needs to be stored in a user-defined OCI object store bucket. Buckets and credentials need to be defined by the user. And those set of backups are user-maintained. The backups can be accessed like any other file on object store.

Staging dump files, Oracle external tables, and other objects used by the database are stored in user-created object storage buckets. And the process for creating and maintaining these buckets is the same as user-defined backup buckets. Oracle Database services are exposed in two different ways. Most of the actions are exposed through the easy-to-use cloud user interface, providing click-through screens for achieving most functions.

Because a lot of database applications are part of a much larger ecosystem controlled through the scripting or other tools, autonomous database provides REST APIs to perform any supported operation. For example, database creation, termination, backup, restore, start, and stop, or scanning CPUs or storage can be performed either through the user interface or the REST APIs. We will be discussing both of these in more detail in future modules.

Autonomous database includes monitoring capabilities available through the Cloud Service dashboard with an easy-to-use UI and can also be performed through Enterprise Manager Cloud controls. Developers and DBAs can use the included SQL developer tool for developing database applications or performing DBA management operations. SQL Developer natively understands how to interface with the autonomous database cloud credentials. So no in-depth knowledge of how to connect to the Oracle Cloud Services is required.

Using Oracle REST Data Services, ORDS, developers can easily build REST APIs for data and procedures in the database. Connecting to the autonomous database is done using credential wallets via SQL*Net, JDBC, or ODBC. The wallet can be downloaded from the Service Console or using REST APIs. We will cover this procedure in more detail in future modules.

Without the wallet and credentials, there is no easy way to access the autonomous database, thus providing a secure, customer-managed process for allowing users to connect in the database. Applications that use JDBC thin driver require the Oracle database credentials, including the Oracle Wallet or Java Key Store, JKS, files when connecting to the autonomous database. The wallet location can be included in a JDBC URL-

- this requires Oracle JDBC thin driver 18.1 or higher-
- or in the ojdbc.properties file, which requires Oracle JDBC thin driver 18.1 or higher as well.

Java properties can be set prior to starting the application. This requires Oracle JDBC thin driver 12.2.0.1 or higher. If you connect to the autonomous database through HTTP proxy, you need to update your tnsnames.ora file to add the HTTP proxy host name and port to the connection definition. In addition, you need to add the https_proxy and the https_proxy_portparameters in the address section of connection definitions.

As with other Oracle Cloud Services, one of the design objectives of the Oracle Autonomous Database is the ability to connect on-premises databases and applications to autonomous databases. Typical scenarios include extract, transform, and load directly into ADB and business intelligence applications accessing ADB. These applications access ADB directly to perform analytics and visualization of data in the database.

The recommended connectivity for autonomous database is through Oracle's FastConnect service, which creates a very fast private network link between the customer's data center and Oracle's Public Cloud. FastConnect Public Peering enables you to access public services in the Oracle Cloud without traffic traversing the internet path. Using FastConnect Public Peering, you can connect to public services like the Oracle Object Storage, public load balancers in your VCN, public IPs on compute, or supported SaaS services, as well as Oracle's Autonomous Database service. FastConnect can be implemented as a co-location or provider model.

For third-party tools accessing Oracle Autonomous Databases-- for example, in this graphic, Cognos-- the recommended connectivity services is through Megaport cloud routers. Megaport makes it easy to connect to Oracle Cloud regions across the US, Europe, and Asia-Pacific. With Megaport, you can provision dedicated and private connections from 386-plus Megaport-enabled data centers to Oracle Cloud Infrastructure, FastConnect and Oracle Cloud Infrastructure, FastConnect Classic in less than 59 seconds. Scalable bandwidth enables you to pay only for what you need, when you need it.

For workloads that have both an application server and an autonomous database in the same Oracle Cloud Infrastructure region-- for example, a web-hosted application on OCI infrastructure-- that accesses data in an autonomous database, access between the two is done through the public IP address of each service. However, this traffic never leaves the OCI region and is not routed through the external public internet. Instead, it is directly routed through a service gateway that connects it to services. This provides higher security, because this traffic will never leave the data center, and provides much lower latency and better bandwidth since all the traffic is routed through the internal high-speed networks in the region.

The Oracle Autonomous Database leverages extended architecture components that enhance its functionality for customer needs. It is integrated, at no cost, with Oracle SQL Developer, which is an extensive development and management tool for the Oracle Database. Also included and integrated into the autonomous database offering is Oracle Machine Learning, which is a notebook-based environment that includes machine learning functionality built into autonomous database.

A third tool included with the Oracle Autonomous Database is Oracle Data Visualization Desktop, which is the extremely capable business intelligence and analytics tool. In the graphic, the components that are included with autonomous database are colored in salmon color. For data movement in and out of autonomous database, the Oracle Cloud Platform provides object storage that the autonomous database uses to stage files that are being loaded into it or for external backups that users want to perform. However, autonomous database uses internal Exadata storage for database object storage and does not require the use of cloud object store for its operation. Many third-party applications are certified against Oracle Autonomous Database and can be connected through OCI, JDBC, and ODBC protocols.

Let's look at the developer tools and some of the other features in the architecture in greater detail. As we previously noted, an included component of the autonomous database service is Oracle Machine Learning, also referred to as OML. OML is a web-based notebook environment based on Apache Zeppelin.

With OML, users can quickly start running queries in an HTML environment without the need to install a client query tool. OML is autonomous database-aware and makes it easy to leverage functionality such as resource services defined, machine learning algorithms in the database, SQL scripts, and graphical analytics tools that are part of OML. Notebooks can be saved and shared with other OML users. And OML provides an easy, integrated SQL and analytic development and runtime environment that can be accessed from anywhere, anytime.

Data Visualization Desktop provides a powerful personal exploration and visualization in a simple, per-user desktop download. Data Visualization Desktop is the perfect tool for quick exploration of sample data from multiple sources or for rapid analysis and investigation of your own local data sets. Data Visualization Desktop makes it easy to visualize your data so you can focus on exploring interesting data patterns. Just upload data files, or connected Oracle applications, or a database, select the elements that you're interested in, and let Data Visualization Desktop find the best way to visualize it.

Choose from a variety of visualizations to look at data in a specific way. Data Visualization Desktop's benefits include a personal single-user desktop application, offline capability, completely private analysis, full control of data source connections, direct access to on-premises data sources, lightweight, single-file download, no remote server infrastructure, and no administration tasks.

Oracle SQL Developer is a free, integrated development environment that simplifies development and management of Oracle Database in both traditional and cloud deployments. SQL Developer offers complete, end-to-end development of your PL/SQL applications, a worksheet for running queries and scripts, a DBA console for managing the database, a report interface, a complete data modeling solution, and the migration platform for moving your third-party databases to Oracle. SQL Developer versions after SQL Developer 17.4 or later can connect to autonomous database using an Oracle Wallet. And this version contains enhancements for key autonomous database features.

Oracle SQL Developer 17.4 and later provide support for wallets using cloud [INAUDIBLE] connection type. Oracle recommends that you use version 18.2 or later however. But earlier versions may still work with autonomous database.

The console overview page shows real-time and historical information about the utilization of the service. The activity page shows real-time and historical information about the utilization of the service. And the administration page allows downloading of client credentials, set resource management rules, set administrative password, manage OML users, download Oracle Instant Client, and provides a mechanism to send feedback to Oracle.

Oracle Autonomous Database is certified with many third-party vendors. And Oracle encourages any vendor that wants to certify their application or tool against the Oracle Autonomous Database to do so. Although autonomous database runs the latest version of the database, it is the exact same version that customers can run on premises or any other cloud service. Oracle restricts some operations against its autonomous services to better control and maintain and to provide true hands-off autonomous experience to users. In this slide, you can see some of the vendors that have certified their applications with Oracle's Autonomous Database.

Now let's go over typical considerations that a customer would evaluate when implementing an autonomous database. When considering a move to, or a new deployment in, autonomous database, what is a typical workflow of planning and deployment that occurs? Unlike on-premises deployments, there are many steps that are not needed with autonomous database. Because of its nature, it is meant to be an easy environment to deploy.

However, there still are several considerations to evaluate-- number one, determining the level of automation and functionality required; number two, determine the main workload characteristics for the database; number three, depending on the workload characteristics, select Autonomous Data Warehouse or Autonomous Transaction Processing service; number four, provision the selected service; number five, determine how to load data into the autonomous database; and number six, determine what to do with the application. Let's examine each of these considerations in more detail.

Oracle Autonomous Transaction Processing supports all operational business systems, including both departmental as well as mission-critical applications. But unlike other cloud providers, ATP doesn't just support one transaction processing use case, it can also support mixed workloads where you have a mixture of transaction processing, reporting, and batch processing, making it the perfect platform for real-time analytics based off operational databases. This enables users to get immediate answers to any question.

Integrated machine learning algorithms make it the perfect platform for applications with real-time predictive capabilities. Advanced SQL and PL/SQL support make it the perfect platform for application developers as developers can instantly create, effortlessly use Autonomous Transaction Processing, eliminating any dependence and delays on others for hardware and software. The fact that it's self-tuning also eliminates any database tuning, accelerates developer productivity.

Oracle Autonomous Data Warehouse supports all types of analytical warehouse and decision support database workloads. ADW is particularly well-suited for creating new dependent or independent data marts that allow easy start of analytical projects. It is a good environment for sandbox experimentation by data scientists sifting through data and for storing large amounts of data and data lakes. Its included analytics and visualization tools, Oracle Machine Learning and Oracle Data Visualization Desktop, provide an end-to-end environment for application development, data analysis, and fast, flexible database services.

Once you decide which service better suits your needs, the next step is to provision the database. Provisioning the database involves very few steps, but it's important to understand the components that are part of the provisioning environment. When provisioning a database, the number of CPUs, in increments of one storage, in increments of 1 terabyte, and backup are automatically provisioned and enabled in the database. In the background, an Oracle is being added to the container database that manages all the users in autonomous databases.

Because the autonomous database runs on Exadata systems, real application clusters is also provisioned in the background to support the on-demand CPU scalability of the service. This is transparent to the user and administrator of the service. But be aware, it is there. For higher-end offerings, there is the option of creating an optional remote standby database for automatic failover.

As mentioned, autonomous database runs on Exadata systems, but no Exadata installation, configuration, or management needs to be done or can be done. init.ora parameters are configured automatically in autonomous database depending on the service selected, ADW or ATP. Memory, parallelism, concurrency, number of sessions, and other parameters are automatically configured based on the number of CPUs allocated to the service. Most of these parameters cannot be modified. And the few that can be modified should only be done for very specific reasons by qualified DBAs. This is discouraged in autonomous database.

Tablespace management is performed automatically by the Oracle Autonomous Database and cannot be changed by the customer. Customers have full access to view the information of the space allocated to their instance, but it cannot be changed. The only input the customer needs to provide is the number of terabytes of data they would like the database to be able to hold. This number can be increased or decreased in real time. And the autonomous database handles adjusting data location based on this user setting.

Loading and maintaining data in the autonomous database can be done as one-time loads best when staged through Oracle Object Store, or as a continuous data ingestion or synchronization with other sources. Autonomous database supports three object stores and can read and write directly to these three. The supported object stores are Oracle Object Store, Amazon S3, and Azure Object Store. Object stores are ideal for staging export dump files that are going to be imported into the autonomous database.

The same applies for flat files that would be loaded into the database. Autonomous database supports the Oracle Database external tables feature. So flat files on object store can act as autonomous external tables. Please note, it is best to host these tables on Oracle Object Stores that are FastConnected to the autonomous database to reduce latency and other issues around access time to database objects.

Also available for transaction and data work location in real or near-real time, or to maintain synchronized copies of the databases, are Oracle GoldenGate, which can be configured with autonomous database as a target database. This allows ADB to become a full replica copy of another database for uses such as reporting, disaster recovery, or development, testing, and QA. Once a decision is made to move the database to autonomous database services, the next step is to determine what to do with the application accessing that database.

Just like rehosting the database in the cloud, re-hosting the application to the cloud may have its own benefits. If the application using the autonomous database is an existing application, there are two preferred options for hosting the application. First option is to keep the application in its existing environment, and replace the existing database with access to the autonomous database. The second option is to rehost the application to the Oracle Cloud Infrastructure. Rehosting the application may be straightforward or may require substantial reconfiguration.

Oracle provides tools such as Ravello to assist in these migrations. If the application using the autonomous database is a new application, then it is highly desirable to also develop the application on the Oracle Cloud Infrastructure Development environment, which will benefit from a robust infrastructure offering and close connectivity to the autonomous database.