|  |  |
| --- | --- |
| Macintosh HD:Users:AMKOTHAR:Pictures:iPhoto Library.photolibrary:Masters:2015:01:11:20150111-122913:IMG_1254.PNG | **Context Setting:**   * With the growing uptake of Platform Services, it is become very cleat that APIs are central to the adoption of PaaS platform. Oracle believe that having a strong API strategy is essential to be a successful Platform Service player * APIs greatly simplify the life of developers and administrators by making it easy to access the functionality of the platform. APIs speeds up the platform access and makes the platform highly extensible leading to a rich feature set. It allows building an ecosystem with other vendors to complete your platform. APIs are the only way to support all kinds of integrations and interoperability. It improves the management of platform security, makes it easy to do deep analytics and even goes a long way to ensure compliance. APIs also allows us to release functionality quickly without the need to build UIs right away, watch how the APIs are used and then design supporting UIs. * With that strategy, Oracle PaaS, especially Java Cloud Service and Database Cloud Service, provide a rich set of APIs to drastically improve your DevOps capabilities. * To illustrate the richness of our PaaS APIs, we have built a tablet application that uses these publicly accessible REST APIs to manage and monitor Java Cloud Service. * One can perform any operation available on the Oracle Cloud Portal. In essence, we have replicated all the functionality on the Cloud Portal, and added little more. * Let’s open this application. This is a native iOS app, built using Oracles MAF technology. It uses PaaS REST APIs to interact with the PaaS Services. |
| Macintosh HD:Users:AMKOTHAR:Pictures:iPhoto Library.photolibrary:Masters:2015:01:11:20150111-122913:IMG_1245.PNG | * Once you login, you see all your subscribed services. In this case, you see Java, Database, and Compute. * You can also see the account balance, some high level stats for each of the subscribed service type, for e.g. you see that there are 2 instances of Java Cloud Service running. You get a bird’s eye view of the aggregate usage across these instances. You get all this information from the account management APIs. * You can create a new instance from here by clicking create instance, which underneath uses the create APIs, or you can simply touch the JCS row and you will get a dashboard listing all Java Cloud Service instances * Oracle PaaS provides you a simple and easy way to instantly create new service instance. Instances can be created by developers or administrators through self-service tooling provided by Oracle or through REST APIs, just on-board the user who needs to privilege to create service instance * Let’s first walk through service instance creation flow |
| Macintosh HD:Users:AMKOTHAR:Pictures:iPhoto Library.photolibrary:Masters:2015:01:11:20150111-122913:IMG_1255.PNG | * If you click on the Create Instance, you will see the following wizard |
| Macintosh HD:Users:AMKOTHAR:Pictures:iPhoto Library.photolibrary:Masters:2015:01:11:20150111-122913:IMG_1256.PNG | * The create wizard walks you through a simple, 4 step flow to provision a service instance * In the first step, you pick the service variation and billing frequency * The wizard also gives an indication on how much your choice will cost you * It uses the create APIs with the inputs from this wizard. Oracle PaaS Platform provides these APIs for both Java and Database services |
| Macintosh HD:Users:AMKOTHAR:Pictures:iPhoto Library.photolibrary:Masters:2015:01:11:20150111-122913:IMG_1257.PNG | * In the next step, select the version and the patch level of the service   NOTE: Version does not impact the price in any way, so there should not be price included on this page |
| Macintosh HD:Users:AMKOTHAR:Pictures:iPhoto Library.photolibrary:Masters:2015:01:11:20150111-122913:IMG_1259.PNG | * Select the compute shape for all the nodes in the cluster of this instance * Same shape is used for all nodes, for e.g. if you pick a cluster size of 4, there will be 4 Virtual Machines with a Managed Server on each one of them. Each of these VM will be of medium size as selected on this page |
| Macintosh HD:Users:AMKOTHAR:Pictures:iPhoto Library.photolibrary:Masters:2015:01:11:20150111-122913:IMG_1260.PNG | * Provide a name, description to the service instance. * Provide the database configuration details to wire * And the storage service for scheduled backups of the environment * Confirm the input and hit provision to create the service * The application calls the create APIs at this point * NOTE: The inputs used in this wizard are based on an old mockup. |
| Macintosh HD:Users:AMKOTHAR:Pictures:iPhoto Library.photolibrary:Masters:2015:01:11:20150111-122913:IMG_1246.PNG | * Let’s get back to the service instance dashboard * The dashboard uses the JCS Account APIs to get overall infrastructure consumption across the 3 instances, for e.g. a total of 228GB of storage is used by these 3 instances, number of service instances in the account and some metadata about the instances, for e.g. the number of nodes, version, etc. * Now let’s see the details of an instance * Click on one of the JCS instances, say “venkyjcs” |
| Macintosh HD:Users:AMKOTHAR:Pictures:iPhoto Library.photolibrary:Masters:2015:01:11:20150111-122913:IMG_1247.PNG | * The app now uses the instance details APIs to get very detailed information about the instance. * The amount of infrastructure consumed by the instance, the size of the cluster (i.e. number of nodes), whether the instance has a web-tier, a load balancer, enabled, etc. * It also provides information on each of the nodes in the cluster – their host information, size of the Virtual Machine, Storage used, etc. All this information is retrieved using the instance APIs * One can also perform actions on these nodes. * For e.g. if we click on the hamburger icon next to the 1st node in the cluster, you will see some actions. |
| Macintosh HD:Users:AMKOTHAR:Pictures:iPhoto Library.photolibrary:Masters:2015:01:11:20150111-122913:IMG_1248.PNG | * You can start/stop the node, you can even jump to other consoles   [Internal Note] This menu option is not correct. These actions don’t belong here. They belong in the instance list page, but not in this instance details page. |
| Macintosh HD:Users:AMKOTHAR:Pictures:iPhoto Library.photolibrary:Masters:2015:01:11:20150111-122913:IMG_1249.PNG | * On the topology tab, one can easily scale the service. Just click on Add Node to add new nodes to the cluster, i.e. scale out. * This function uses the JCS PaaS Scale APIs. * The hamburger icon next to each node allows you to remove the node from the cluster, i.e. scale in. * All these APIs are publicly available for you to use with any 3rd party tool, or build custom tooling on your own. They are designed to simplify your DevOps process within your company. * Similar to Scale, there are instance level APIs to backup, restore, patch, rollback a patch, etc. Every lifecycle operation has a corresponding API that you can directly use instead of using Oracle provided service consoles. * Many of the REST APIs are more capable than the out-of-the-box service console capabilities, for e.g. the default tooling allows adding a new node to the default cluster in the WebLogic domain. But through REST API one can target adding new nodes to a different cluster. * Cloud tooling focuses on the most common functions to keep tooling simple and easy to use, while REST APIs expose all the advanced features |
| Macintosh HD:Users:AMKOTHAR:Pictures:iPhoto Library.photolibrary:Masters:2015:01:11:20150111-122913:IMG_1250.PNG | * Scaling out is as simple as adding specifying the number of nodes to be added * Few minutes later the service is scaled up and you will see the new topology with added capacity |
| Macintosh HD:Users:AMKOTHAR:Pictures:iPhoto Library.photolibrary:Masters:2015:01:11:20150111-122913:IMG_1251.PNG | * REST APIs are available for all life cycle operations. * In the administration tab, you can see the options to patch, backup, etc. * At the instance level, there are APIs to list available patches, patch history, etc. * If there is any applicable patch, you will see a “Patch Now” button. It allows you to apply the patch to the entire service. The application calls the patching API to execute the patch operation. |
| Macintosh HD:Users:AMKOTHAR:Pictures:iPhoto Library.photolibrary:Masters:2015:01:11:20150111-122913:IMG_1271.PNG | * If you jump on the backup tab, you will see a list of backups – scheduled and on-demand backups. The APIs also provide backup history, app uses this API to list all the backups * You can change the default configuration of the backups, like the time of day to perform the backup, etc. |
| Macintosh HD:Users:AMKOTHAR:Pictures:iPhoto Library.photolibrary:Masters:2015:01:11:20150111-122913:IMG_1270.PNG | * If you are about to make a change and would like to take a backup in case something goes wrong, click on “Backup Now” to perform an on-demand backup * You can choose a full or incremental backup * The app calls the instance backup APIs with the inputs from this action * The APIs also provide backup history * You can restore the service from a prior backup at any time. Click on Restore Now, and the service will be restored at that point in time. The app calls the instance restore APIs to perform the action. |