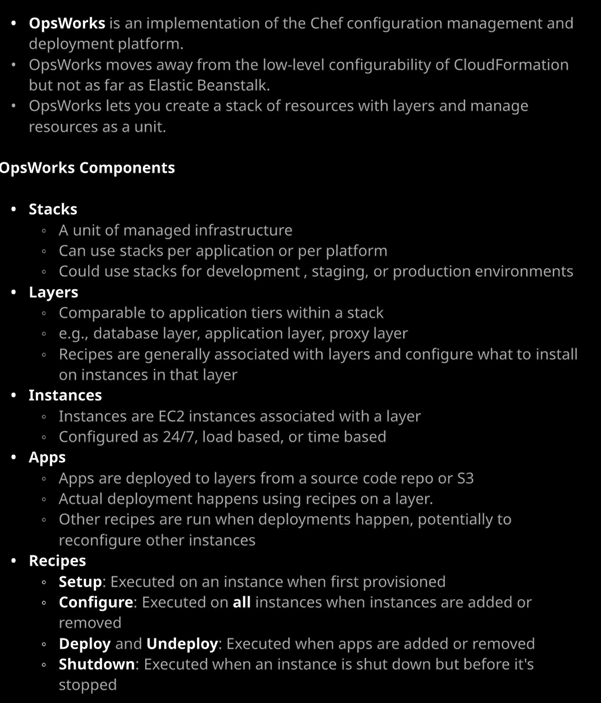
Welcome back and in the last lesson I talked about Elastic Beanstalk.

So **OpsWorks is an infrastructure management product available within AWS, which allows you to manage large sets of infrastructure. It takes away a lot of the admin overhead, but it still gives you a lot of the flexibility that you get from CloudFormation a lot of the automation that you get from Elastic Beanstalk but it is designed for an infrastructure engineer**. **It's not designed for a developer**. **OpsWorks is based on the Chef configuration management platform, and if you don't want that is, I'll include some links in the lesson description but essentially, OpsWorks functions with a number of these base components, and I want to step through these one by one.**



OpsWorks is not the type of thing that I can demonstrate how to use in a lesson, because a lot of the power comes from managing large scale infrastructure but what I can do is step through the architecture of how OpsWorks operates and give you a little bit of an idea about some of the questions that you might face in the exam, so I'm just going to go ahead and move across the OpsWorks part the console. So OpsWorks allows you to create a stack of resources. So, just like CloudFormation uses the concept of a stack, OpsWorks also uses a stack as its base entity, but it's not a stack in exactly the same way, so both of them represent collections of infrastructure but with CloudFormation, a stack is created from a CloudFormation template. With OpsWorks, it's just an entity that you configure and then begin to build on that, using these other configuration components. So let's get started. I'm going to create a stack inside OpsWorks. Now you can create a stack based on either Chef 11 or Chef 12. **So with Chef 11, it's capable of managing Linux based operating systems. With Chef 12, you get to use Linux or Windows**, so that's an important distinction to keep in mind and it is likely something that will come up in the exam if you do have any OpsWorks questions, I'm going to go ahead and create a stack that's based on Linux. So I select that and go to create Stack. Now OpsWorks does use IAM and so it will need to create some permissions in order to interact with different components of AWS, so we'll go ahead, create the stack. Now OpsWorks is based on Chef, Chef uses recipes, recipes use cookbooks, and cookbooks require a repository. So all that needs to be created, and I'll talk about exactly what that means in a second but now I'm going to click on explore the sample stack because you just want to draw your attention to a number of the important elements for the exam. So within OpsWorks, a stack is a unit of managed infrastructure. When you're controlling permissions and giving people permissions to interact with OpsWorks, you generally doing it on a per stack basis. Now you can create stacks and have a stack per application. You can have a stack per platform. You could use one stack for development, one stack for staging, one stack for production. Your exact structure of how you utilize these stacks depends on exactly how you want to implement your platform, but essentially a stack is an isolated grouping of infrastructure. Now, within a stack, you've got one or more layers, and you can compare layers to tiers within an application, you might have a web server layer. You might have an application layer. Maybe you've got a proxy layer. You might have a database layer. The key thing about these layers is it's on the layers that you can specify to apply recipes and recipes are essentially the documents that Chef uses to configure the instances that are inside these layers. So a stack has one or more layers, and inside the layer is where your instances are, where the compute services are that will actually do the work as part of your stack. So with a layer you can add instances. So in this particular case, this layer has one instance, and it's currently in the stopped state but I can click on start to go ahead and start this instance. Now instances within layers are added a number of different ways. They could be configured to run 24/7 so you can just add an instance that runs for the entire life span of the stack in the layer in this particular case this is one of those instance types, and I can manually stop and start it but if it is started its running consistently. We've also got time based instances and time based instances can be set to start and stop based on a specific schedule. We've also got load based instances and with load based, we can configure OpsWorks to automatically scale in and out the number of instances based on the load that's incoming to that layer. So we've got a full set of scaling options, and all of those are configured on a per layer basis. Layer is also where we define the recipes and because it's based on Chef the recipes are the actual things that do the configuration, so a layer has a number of different recipe types. It might be set up to have a setup recipe, a configure recipe, a deploy recipe, and undeploy recipe and shutdown recipe. So these serve different functions. The setup recipe is what's executed on an instance when it's first provisioned so this might perform certain tasks, like installing a web server or installing base system components. The configure recipe is executed on all instances of that layer whenever you add or remove an instance to that layer and an example of why that might be used if you've got a cluster running inside that layer and you add an instance, you want all of the other instances in that layer to be aware of the newly added instance, or the newly removed instance and so, by having this configure recipe evaluate what instances remain in that layer can update the configuration of all of the other instances. Now the deploy and undeploy recipes are executed whenever applications are added or removed to that layer. The shutdown recipe is executed whenever an instance is shut down book before it's stopped. So these recipes really are the core of the Chef infrastructure management platform and based on the fact that OpsWorks users Chef, the recipes are the core functionality of OpsWorks and they're defined on a layer by layer basis. So we've got the stack, which is the base level component. A stack has one or more or layers and if we go to layers and then go to add layer we can either add an OpsWorks layer, an ECS layer, so if you've got an ECS cluster that is already present inside your AWS account, then you can add that and use that functionality inside OpsWorks. We've also got a RDS layer. So if you have any RDS database instances than they can be referenced as a layer inside an OpsWorks stack. So you got the different types of layers you can add to a stack and then inside those layers, we've got instances you've got different configuration settings that you can control on a layer by layer basis just to give you some examples of exactly what we can do, we've got auto healing settings. We can define the exact network configuration of any instances that are added to the layer. You can control the storage of the instances that are added to that layer. So if you wanted to add additional mount points to the instances that get added, you could go ahead and do that. The security can be configured on a layer by layer basis. You can configure CloudWatch log exports on a layer by layer basis, and then, obviously, to those instances using the recipes you've got the ability to deploy applications and an application, as the name suggests, is simply an application that you deploy to one or more layers within a stack. Now the application itself is deployed either from a source code repository, or S3, and the actual deployment of the application is using the recipes that are associated with that layer. Now the exact way that Chef works that is well beyond the scope of what you need for this course and probably OpsWorks as a product is beyond what you need, strictly speaking for the exam but as always, I want you to be aware of exactly how a product functions, even if that's just at a high level. What I will do is make sure include a good selection of links in the lesson description, which will give you some surrounding reading around OpsWorks and Chef, if it's an area that you feel you want to know more on but with that being said, that is everything that I wanted to cover in this lesson. This is the last lesson of this topic, and it is the last content topic and content section of the course. So now, at this point, you should have done all of the content lessons in the course. So now we need to start focusing on getting you a really good exam pass and so for the rest of the course, I want to focus on exactly what things you need to focus on for the exam. So I have a lesson coming up, which will detail my thoughts on exactly what you need to do and expect for the exam. I'll include some links for additional background reading, some links to YouTube videos from additional deep dive topics that I think will be valuable for you to watch before the exam and and then, of course, there'll be a full practice exam with some exam quality questions. So at this point, go ahead, mark this lesson as complete, and when you're ready, you can join me in the next.